Managing Flash Communication Server

Macromedia Flash™ Communication Server MX 1.5

Trademarks

Afterburner, AppletAce, Attain, Attain Enterprise Learning System, Attain Essentials, Attain Objects for Dreamweaver, Authorware, Authorware Attain, Authorware Interactive Studio, Authorware Star, Authorware Synergy, Backstage, Backstage Designer, Backstage Desktop Studio, Backstage Enterprise Studio, Backstage Internet Studio, Contribute, Design in Motion, Director, Director Multimedia Studio, Doc Around the Clock, Dreamweaver, Dreamweaver Attain, Drumbeat, Drumbeat 2000, Extreme 3D, Fireworks, Flash, Fontographer, FreeHand, FreeHand Graphics Studio, Generator, Generator Developer's Studio, Generator Dynamic Graphics Server, Knowledge Objects, Knowledge Stream, Knowledge Track, Lingo, Live Effects, Macromedia, Macromedia M Logo & Design, Macromedia Contribute, Macromedia Flash, Macromedia Xres, Macromind, Macromind Action, MAGIC, Mediamaker, Object Authoring, Power Applets, Priority Access, Roundtrip HTML, Scriptlets, SoundEdit, ShockRave, Shockmachine, Shockwave, Shockwave Remote, Shockwave Internet Studio, Showcase, Tools to Power Your Ideas, Universal Media, Virtuoso, Web Design 101, Whirlwind and Xtra are trademarks of Macromedia, Inc. and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words or phrases mentioned within this publication may be trademarks, servicemarks, or tradenames of Macromedia, Inc. or other entities and may be registered in certain jurisdictions including internationally.

Third-Party Information

Speech compression and decompression technology licensed from Nellymoser, Inc. (www.nellymoser.com).

Sorenson™ Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

This guide contains links to third-party websites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Macromedia provides these links only as a convenience, and the inclusion of the link does not imply that Macromedia endorses or accepts any responsibility for the content on those third-party sites.

Copyright © 2002-2003 Macromedia, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.

Acknowledgments

Director: Erick Vera

Producer: JuLee Burdekin

Writing: Jay Armstrong, Jody Bleyle, JuLee Burdekin, Barbara Herbert, Barbara Nelson, Shimul Rahim

Managing Editor: Rosana Francescato

Editing: Mary Ferguson, Anne Szabla

Production Manager: Patrice O'Neill

Multimedia Design and Production: Aaron Begley, Benjamin Salles

Print Design, Production, and Illustrations: Chris Basmajian

Second Edition: March 2003

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

CONTENTS

INTRODUCTION

About This Manual 5

Intended audience 5
System requirements 5
About the Flash Communication Server documentation 6
Typographical conventions 6
Additional resources 6

CHAPTER 1

About your installation 7

About client-side and server-side files 7
About the installed files 8

CHAPTER 2

Managing the Server 11

Performing administration tasks 11
Using the Administration Console 16
Performing advanced administration tasks in Windows 23
Managing Flash Communication Server on UNIX 24

CHAPTER 3

Configuring Flash Communication Server 27

Typical configurations 27
About configuration levels 31
About the configuration files 34

CHAPTER 4

Understanding Flash Communication Server Security 51

Managing server security 51
About authentication and authorization 53
Choosing passwords 54
Developing secure applications 54
About privacy 55
Deploying secure applications 55

INDEX 57

INTRODUCTION
About This Manual

Macromedia Flash Communication Server MX 1.5 enables one-to-one, one-to-many, many-to-one, and many-to-many communication in real time between applications created in Macromedia Flash MX. Developers create these applications using ActionScript, a scripting language based on the same standard used by the JavaScript language.

Flash Communication Server MX communicates with the Flash Player using the Real-Time Messaging Protocol (RTMP), an unencrypted TCP/IP protocol designed for high-speed transmission of audio, video, and data messages. Because RTMP packets often aren't allowed through a firewall, Flash Communication Server can also wrap RTMP packets in HTTP, using a protocol called RTMPT (tunnelled RTMP).

This manual describes how to configure and manage Flash Communication Server to support communication applications deployed on a variety of network configurations. The rest of this chapter provides system requirements, describes the Flash Communication Server documentation, and lists additional resources.

Flash Communication Server enables applications to communicate with other servers. This manual does not discuss web server and application server management or server operating system setup.

Intended audience

This manual is aimed at system administrators who will configure and manage Flash Communication Server to support communication applications. You should already be familiar with basic network infrastructure and security. You should also have some familiarity with client-server application models, XML, and JavaScript.

System requirements

Flash Communication Server can be deployed on the following systems:

- Microsoft Windows 2000 Server or Windows NT 4.0 Server (SP6 or later), running on a Pentium III 500 MHz processor or greater (dual Pentium 4 or better recommended). Windows XP is acceptable for developing and testing applications.
- Red Hat Linux 7.3 or 8.0 running on a Pentium III 500 MHz processor or greater (dual Pentium 4 or better recommended).

Your deployment system for Flash Communication server applications also requires a minimum of 256 MB of available RAM (512 MB recommended); 50 MB of available disk space; and a CD-ROM drive for installation. Depending on your applications, your requirements might be greater than outlined here.

If you install Flash Communication Server on a UNIX computer, you'll need Flash MX installed on a separate Windows or Macintosh computer to develop your Flash applications. You'll also need Macromedia Flash Player for Windows or Macintosh and a web browser to run the sample applications and administration and help tools provided with Flash Communication Server. Download the Flash Player from the Macromedia website (www.macromedia.com).

For the best experience with Flash Communication Server Help, Macromedia strongly recommends that you use a browser with Java player support, such as Microsoft Internet Explorer 6.0 or later. Flash Communication Server Help also supports Netscape 6.1 or later on Windows and Macintosh. Running Flash MX and Flash Communication Server Help simultaneously on a Macintosh may require up to 32 MB of memory, depending on your browser's memory needs.

About the Flash Communication Server documentation

All Flash Communication Server documents are available in PDF format (viewable and printable with Adobe Acrobat Reader) and as HTML help. The PDFs are available on the product CD or can be downloaded from the Macromedia website. To view the HTML help, open the Welcome page (in Windows, Start > Programs > Macromedia > Flash Communication Server MX > Welcome) or in Flash, select Help > Welcome to FlashCom.

You can access tutorials on how to build simple applications that demonstrate single, important concepts. These tutorials are located in the Flash Communication Server Welcome page.

Typographical conventions

The following typographical conventions are used in this book:

- `Code font` indicates ActionScript statements, XML tag and attribute names, and literal text used in examples.
- *Italic* indicates placeholder elements in code or paths. For example, \settings*myPrinter*\ means that you should specify your own location for *myPrinter*.
- Directory paths are written with backslashes (\) for servers running on Microsoft Windows systems. If you are running Flash Communication Server on a UNIX system, replace the backslashes with forward slashes (/).

Additional resources

The Flash Communication Server documentation was written before the code in the product was complete. Therefore, there may be discrepancies between the final implementation of the product's features and how they are documented in this manual. For a list of known discrepancies, see the documentation update in the Flash Support Center (www.macromedia.com/go/flashcom_documentation_update).

The Flash Support Center is updated regularly with the latest information on Flash and Flash Communication Server, as well as advice from expert users, advanced topics, examples, tips, and other updates.

CHAPTER 1
About your installation

This chapter describes the installation directories and the files that are installed during the Macromedia Flash Communication Server MX 1.5 installation process.

For information on how to install the server and the authoring extensions, see *Installing Flash Communication Server*.

About client-side and server-side files

During installation, you determine where your client-side application files are installed. Server-side application files are installed in a default location. You should understand the difference between these files:

- Client-side application files are SWF and HTML files. The installer installs SWF and HTML files for the Macromedia Flash Communication Server MX 1.5 sample applications in either the web publishing directory of your choice or the installation directory. However, when you begin developing applications, you can store the client-side files for your applications anywhere.
- Server-side application files include script files (ASC files), stream files (such as FLVs), and shared object files (FSOs) and must always remain on the server computer. By default, the installer installs server-side files for the sample applications in C:\Program Files\Macromedia\Flash Communication Server MX\applications in Windows, and /opt/macromedia/fcs/applications on UNIX. (This directory is referred to as the applications directory.) The location of the applications directory is written to the `<AppsDir>` tag in the Vhost.xml file. The server will therefore always look for server-side application files in the location specified in `<AppsDir>`.

  When you begin developing applications, you must create a registered application directory for your application in the applications directory with the name of the application and keep the application's server-side files there. For example, for your application called my_app, create a my_app subdirectory in the applications directory and put the server-side files for my_app in that subdirectory (.../applications/myapp).

During development, you can keep client-side and server-side application files together, if you want. When you deploy applications on a public server, you'll need to separate client files from your server-side source files. Your SWF and HTML files should be accessible through a web server, if you're using a web server. Your server-side ASC files, your audio/video FLV files, FSO files, and your ActionScript FLA source files should not be accessible to a user browsing your website.

About the installed files

When the server has been installed, it's a good idea to become familiar with the files placed in the Flash Communication Server directory and, if applicable, your web root directory.

Files in the installation directory

If you chose the default directory in the installer, the Flash Communication Server MX installation directory is located at C:\Program Files\Macromedia\Flash Communication Server MX\ (Windows) or /opt/macromedia/fcs/ (UNIX).

The Flash Communication Server directory contains these items:

- FlashCom.exe (Windows) or fcserver (UNIX) is the server application.
- FlashComAdmin.exe (Windows) or fcadmin (UNIX) is the server's administration controller. When administrators connect to the server with the Administration Console, they are actually connected to the Admin Service, which communicates with the server to perform administration tasks.
- The tools directory contains BAT files that can be used to start and stop the server (Windows only).

  The tools directory contains scripts used to manage the server and the administration services (UNIX only).
- License.htm (Windows) and license.txt (UNIX) contain the text of the Flash Communication Server license agreement.
- Js32.dll contains the server's Server-Side Communication ActionScript engine (Windows only).
- TcSrvMsg.dll, xmlparse.dll, and xmltok.dll are additional components of the server (Windows only).
- tcSrvMsg is a component of the server (UNIX only).
- The conf directory contains the server's hierarchy of XML configuration files.
- The applications directory contains registered application directories for the sample applications, which you can use to experiment with the server.

  The applications directory is also the directory in which you create registered application directories for your communication applications. For example, create a my_app subdirectory in the applications directory; put your server-side application files for the my_app application in that subdirectory.

  At installation time, the location of the applications directory is written to the `<AppsDir>` tag in the Vhost.xml configuration file.
- The samples directory contains SWF and HTML files for the sample applications. (If specified during installation, this directory might be installed in your web root directory instead.)
- The flashcom_help directory contains documents related to help, including FLA files described in *Developing Communication Applications*.

- Within the flashcom_help directory is the admin directory. This directory contains the Administration Console (admin.swf) and the Communication App Inspector (app_inspector.swf), both of which can be used remotely to connect to the server. You use the Administration Console to monitor and control server activity. You use the Communication App Inspector to monitor and control running applications.
- The scriptlib directory contains the ASC files for the communication components and NetServices (used with Macromedia Flash Remoting).

  At installation time, the location of the scriptlib directory is written to the `<ScriptLibPath>` tag in the Application.xml file.
- (Windows only) If you have Macromedia Flash MX installed on the same computer, the Flash Communication Server installer adds the new Communication App inspector and the NetConnection Debugger windows to Flash MX. These features provide support for creating and debugging communication applications within Flash MX. For more information on using these features, see *Developing Communication Applications.*

If you installed the server on a UNIX system, install the authoring components installer on your Flash MX computer to get the Communication App inspector and the NetConnection Debugger menu options in Flash MX, along with the communication components and the Flash Communication Server help.

Files in the web root directory (optional)

If you have a web server installed and you chose to install the Flash Communication Server sample applications there, you can find SWF and HTML files for the sample applications in a subdirectory of the web root directory. (If you didn't specify a web root directory, the SWF and HTML files are installed in a subdirectory of the installation directory.)

CHAPTER 2
Managing the Server

As a Flash Communication Server administrator, you'll need to perform several administrative tasks after the server is installed. This chapter describes how Macromedia Flash Communication Server MX 1.5 is configured when you first install it, how to set up additional administrators, and how to monitor the server's activity.

For many of these tasks, you'll use the Administration Console that was installed with the server. This chapter describes the Administration Console in detail.

Performing administration tasks

When you've installed Flash Communication Server, you should perform the following tasks before connecting to the server with client applications:

- Review the included sample applications.
- Become familiar with the server's initial configuration.
- Define additional server administrators.
- Register client applications.
- Configure virtual hosts.
- Configure new applications, including deploying server-side scripts or audio/video files.
- Become familiar with the procedures for starting and stopping the server.

Reviewing the sample applications

When you install Flash Communication Server, you can choose to install the sample applications. These applications are designed to show you the functionality of Flash Communication Server and also provide templates for your own work.

To see the samples, in Windows, select Start > Programs > Macromedia > Flash Communication Server MX > Welcome; then choose Sample Applications. On UNIX systems, open a sample SWF file from a computer that has Flash Player installed. The SWF files for the samples are located in the samples directory in either the web server publishing directory or the installation directory.

Understanding application elements

It is important to understand the files, scripts, and other parts of a communication application that runs on the server. These elements include the following:

A Macromedia Flash MX SWF file. This is the user interface to the application. The SWF file can be served from a web server or even distributed to users in other ways, such as e-mail.

A directory inside the applications directory, named to match the application name. This is the *registered application directory* and is what the SWF file uses when connecting to Flash Communication Server. For more information about the applications directory, see Registering client applications.

An optional Application.xml file in your application directory in the applications directory in the Flash Communication Server directory. If present, this file provides specific settings for the application that may be different from the settings in the server's primary Application.xml file. For more information about the server's configuration files, see About the configuration files.

Optional server-side scripts. Most communication applications will need Server-Side Communication ActionScript. If you use server-side scripts, put them in the registered application directory in the applications directory or in the directory specified in the `<ScriptLibPath>` tag in the application's optional Application.xml file. These scripts may have file extensions of .js or .asc.

Optional audio and/or video stream files. Some applications may make use of preexisting audio/video streams or may record them to disk. Stream files have the extension .flv or .mp3 and are located in the /streams/*instance_name* directory within the registered application directory in the /applications directory or in the directory specified in the `<StreamManager>:<StorageDir>` tag in the application's optional Application.xml file.

Optional shared object files. Some applications may make use of preexisting shared objects or may write them to disk. Shared objects contain nonstreaming data that is used by more than one client of a communication application. Shared object files have the extension .fso and are located in the /sharedObjects/*instance_name* directory within the registered application directory in the applications directory or in the directory specified in the `<SharedObjManager>:<StorageDir>` tag in the application's optional Application.xml file. Shared objects can also exist on the client side. For more information about using shared objects, see *Developing Communication Applications*.

Tip: Macromedia recommends you use all lowercase letters and no spaces when naming directories and files used with Flash Communication Server and its applications. This practice helps to ensure that, during development, your applications will work if you move files to different computers on different platforms.

Understanding basic server settings

When Flash Communication Server is first installed, it's configured in a generic way so that you can begin using it with the sample client applications. You should become familiar with this configuration so that you can make decisions about how to change it to suit your needs.

The server is installed with a set of configuration files in XML format. These files define a default server adaptor, a default applications directory, default server administrators, and default settings for application behavior.

The default server adaptor uses port 1935, the number assigned to Flash Communication Server by the Internet Assigned Numbers Authority (www.iana.org). Although you can use any port number, this increases the risk of conflicting with another application that may be assigned to the same port; for example, if you configure the server to use port 80 to support HTTP tunneling, the server might not run both a web server and the Flash Communication Server bound to port 80. Applications must be authored to connect to the same port the server is using, in the NetConnection.connect statement. Be sure the port is not blocked by a firewall.

The server is preconfigured with one adaptor containing one virtual host. The virtual host is equivalent to a domain name. The default applications directory for the default virtual host is the the applications directory in the Flash Communication Server directory. You can view this location by looking at the value for the `<AppsDir>` tag in the vhost.xml file. This directory is where the server will look for application subdirectories at startup; you must place an application subdirectory here for each client application that you plan to connect to the server, and the client subdirectory must have the same name as the client application. The presence of the application subdirectory registers the application with the server.

If you installed the sample applications, the applications directory includes sample applications that are provided to illustrate the essential capabilities of the server. The server-side files for each sample application reside in its self-named directory in the applications directory. The SWF files for the sample applications are located in the samples subdirectory of either the web publishing directory or the installation directory.

You can add adaptors and virtual hosts and change the location of the applications directory by editing the server's configuration files and creating directories in the server's conf directory. For more information, see Chapter 3, "Configuring Flash Communication Server," on page 27.

The default server administrator has the user name and password you chose during the Flash Communication Server installation, and is defined in the Server.xml configuration file. The server administrator can connect to the Admin service with the Administration Console and perform a variety of server administration tasks, including shutting down the server and disconnecting client applications. (In the nomenclature of server administration, this server administrator is equivalent to the "superuser.")

Virtual host administrators can only perform tasks relating to the applications running on their own virtual host. There are no virtual host administrators defined when the server is first installed. Server administrators, including the default server administrator defined during installation, have access to all virtual hosts. Server administrators can add or delete virtual host administrators using the Administration Console. For more information about adding administrators, see Adding and editing administrators.

Registering client applications

The server is configured at installation with one adaptor directory named _defaultRoot_ containing one virtual host directory named _defaultVHost_. The server defines its virtual hosts at startup by searching for directories within the adaptor directory that contain valid Vhost.xml files, such as the _defaultVHost_ directory. At the same time, the server defines each application that will be allowed to connect to a virtual host by looking for application directories inside a directory specified by the `<AppsDir>` tag in the Vhost.xml file (by default, the applications directory in the Flash Communication Server directory).

For example, if you create an application named my_app, you must create a my_app subdirectory in the applications directory to register my_app.

You can specify the directory you want to use to store your client application directories by editing the <AppsDir> tag in the Vhost.xml file. By changing the path specified in this tag, you can locate the directory for the virtual host anywhere you want. If no directory is specified, it defaults to the virtual host directory itself.

To edit the <AppsDir> tag in the Vhost.xml file:

1. Locate the Vhost.xml file for the virtual host you are working with.
2. Open the file in a text editor.
3. Replace the path inside the <AppsDir> tag with the path of your choice, such as C:\Server Files\applications. Do not use quotation marks. To specify multiple directories, delimit each directory path with a semicolon.
4. Save the Vhost.xml file.

You must restart the server in order for this change to take effect.

Once you have specified the directory where you'll store your application directories, you must create a directory inside it for each client application you plan to use with that virtual host. Each client application must have a directory with the same name that the client application uses when connecting to the server. Once you have created a subdirectory for each of your applications, you can decide whether to give any of the applications their own Application.xml file. By including an Application.xml file in a client application's directory, you can give that application different settings from those defined in the virtual host's Application.xml file, which serve as the default settings for applications on the virtual host. For more information about the Application.xml file, see Chapter 3, "Configuring Flash Communication Server," on page 27.

Configuring virtual hosts

With some editions of the server, you can add virtual hosts to the server's configuration. This is useful for separating sets of applications and allows you to define administrators who have access only to a specific virtual host. It is also useful if you are an Internet service provider who is hosting websites using Flash Communication Server.

Each virtual host on the server is associated with an adaptor.

To create a new virtual host, create a new virtual host directory inside the /conf/*adaptor_name* directory in the Flash Communication Server directory, for the adaptor you want to use for the new virtual host:
/conf/*adaptor_name*/*virtual_host_name*. This directory must include the following items:

- A Vhost.xml file
- An Application.xml file

You also need to add the new virtual host to the Server.xml file and add administrative users for that virtual host.

Deploying server-side scripts

In developing applications for Flash Communication Server, you may decide to use server-side scripts to implement some of the functionality. To deploy server-side scripts, you can store them in your registered application directory for the application that uses them or in a scripts directory (which you create) within your application directory. Server-side scripts should always reside on the computer where Flash Communication Server is installed.

For example, you could store the main.asc file for my_app in the following directory: ...*installation_directory*/applications/my_app/scripts.

If you create server-side scripts that use characters that are not in the classic 8-bit ASCII character set, such as non-English characters, you must use a text editor that encodes text in UTF-8 format. Macromedia Dreamweaver MX can encode text in this format.

Script files that are encoded in UTF-8 format must be transferred to the server via a binary file transfer.

For more information about using server-side scripts, see *Developing Communication Applications.*

The Communication App inspector and NetConnection Debugger

If you have Flash MX installed, you'll find two new menu items related to Flash Communication Server: the Communication App inspector and the NetConnection Debugger.

- The Communication App inspector lets you view information about the applications running on the Flash Communication Server.
- The NetConnection Debugger lets you view information about the events that are taking place on the server. These events include client application connection requests and initiation of data streams. This information can be useful when you are testing and debugging your Flash applications.

For detailed information about using these windows, see the "Debugging and Monitoring Applications" chapter in *Developing Communication Applications*.

Starting and stopping the server in Windows

In Windows, Flash Communication Server runs as a service; it does not appear in the Windows taskbar. Therefore, you don't shut down or restart the server as you would other Windows applications.

You can shut down and restart the server in a few ways:

- Use the Administration Console to connect to the server and then shut it down or halt it remotely. (Only server administrators can perform these tasks.) For more information, see Using the Administration Console and Using the Maintenance panel.
- Use the BAT files in the Flash Communication Server MX/tools directory.
- Use the Windows Services control panel.

To start or stop the server by means of the BAT files:

In the Start menu, choose Programs > Macromedia > Flash Communication Server MX > Start Service or Stop Service. The BAT file executes and then closes automatically.

To stop the server in the Services control panel:

1. In the Windows Start menu, select Settings > Control Panel.
2. In the Control Panels folder, double-click the Administrative Tools folder.
3. In the Administrative Tools folder, open the Services control panel.
4. In the Services list, scroll down and select Flash Communication Server.
5. Click the Stop button at the top of the control panel. The server shuts down.

To restart the server in the Services control panel:

1. Open the Services control panel.
2. Select Flash Communication Server.
3. Click the Start button at the top of the control panel.

   The server starts up.

Starting and stopping the server on UNIX

On UNIX systems, Flash Communication Server is installed as a service. You start and stop the Flash Communication Server service using the fcsmgr utility.

To start the server on UNIX:

1. Log in as a root user.
2. Change to the directory where the server is installed.
3. Open a shell window and type:

```
fcsmgr server start
```

To stop the server on UNIX:

1. Log in as a root user.
2. Change to the directory where the server is installed.
3. Open a shell window and type:

```
fcsmgr server stop
```

Use the fcsmgr utility to perform other tasks as well, such as configuring the service to start automatically when the system is started. For more information on this and other tasks, see Using the fcsmgr utility.

Using the Administration Console

You can do common server administration and monitoring tasks using the graphical user interface of the Administration Console and, in Flash MX, the Communication App inspector. These are Flash applications that Macromedia created using public APIs (application programming interfaces). The source files for these applications are available to registered users on Macromedia's Flash MX Support Center (www.macromedia.com/support/flash/). You can use the same ActionScript APIs to create your own custom administration tools or extend the ones provided with the server.

When you install Flash Communication Server in Windows and on UNIX systems, the installer places the Administration Console (admin.swf) in the admin directory. If you have the Flash Player installed, you can monitor and control the server's activity by launching the Administration Console and connecting to the server.

To run the Administration Console from a computer other than the computer where the server is installed, you can either copy the admin directory to the other computer; or, you can ensure that the admin directory is located in a web server publishing directory that can be accessed remotely. In either case, you'll need to ensure the `<Allow>` and `<Deny>` tags in the Server.xml file allow connections from the other computer's IP address. For more information, see About the configuration files.

With the Administration Console, you can perform the following tasks:

- Check the status of the server and the applications running on it
- Shut down or restart the server, a virtual host, or individual client applications
- Add and edit administrators
- View logs of server connections and other server events
- View and update the server's license key and its bandwidth and connection limits

As a security feature, when you connect to the server with the Administration Console, the console actually connects to a separate Admin service that runs in parallel with the server service. The Admin service then communicates with the server to perform its administration functions. On UNIX, you must first explicitly start the Admin service using the fcsmgr utility. Once the Admin service is started, authorized administrators can access the Administration Console.

To start the Admin service on a UNIX system:

1. Log in as a root user.
2. Change to the directory where the server is installed.
3. Open a shell window and type:

```
fcsmgr adminserver start
```

For more information on fcsmgr commands, see Using the fcsmgr utility.

To connect to the Flash Communication Server with the Administration Console:

1. In Windows, from the Windows Start menu, select Programs > Macromedia > Flash Communication Server MX > Administration Console, or open the admin.swf file in a web browser on a computer with the Flash Player installed.
2. In the Host text box, enter the address of the server you want to connect to.

   The default address is localhost, which refers to the computer that the Administration Console is running on. If you are connecting remotely by running the Administration Console on another computer, enter the address of the server you want to connect to, such as FlashComServer.myCompany.com or 12.34.56.78. If the `<HostPort>` tag in the Server.xml file is set to a port other than 1111, include the port number after the host name, separated by a colon. (For more information about server configuration, see Chapter 3, "Configuring Flash Communication Server," on page 27.)
3. In the Name and Password boxes, enter the name and password you entered during the Flash Communication Server installation. If you've changed the administrator user name and password using the Administration Console or manually in the Server.xml file, enter the new user name and password.

   When logging on to a virtual host that is not on the default adaptor, virtual host administrators must specify the name of the adaptor. For example, if a virtual host administrator is logging on to a virtual host on the adaptor `_secondAdaptor_`, the administrator `JLee` would enter the following information in the Name box: `_secondAdaptor_/JLee`.
4. If you want the Administration Console to remember your login and password when you use it in the future, select the Remember Connection Data option.

5 If you want the Administration Console to automatically connect to the server when you open it, select the Automatically Log In option.

6 Click the Connect button.

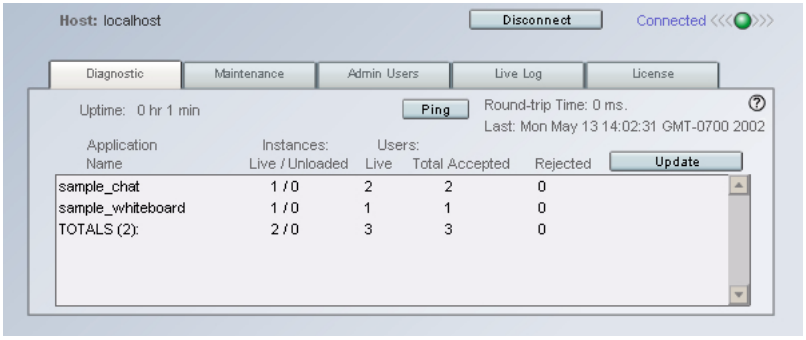
You are now connected to the server, and the Diagnostics, Maintenance, Admin Users, Live Log, and License panels appear. These panels perform the following functions:

- The Diagnostics panel displays information about the applications that are connected to the server and the number of instances and users of each one.
- The Maintenance panel allows you to shut down the server, virtual hosts, or an individual application connected to the server.
- The Admin Users panel allows you to add and edit administrators' log-on information.
- The Live Log panel lets you view information about connections, disconnections, and other server events as they happen.
- The License panel displays the server serial number and information about the number of connections and bandwidth enabled by that serial number.

Tip: If you installed Flash Communication Server on a UNIX computer but did not enter the serial number during installation, the serial number is not displayed. You can enter it using the License panel. (See Viewing and editing license information.)

Viewing server diagnostics

To view information about the server's responsiveness, the amount of time the server has been running, and the applications that are connected to the server, you use the Diagnostics panel.



The Diagnostics panel includes the following items:

Uptime indicates how long the server has been running continuously and is updated once per minute. The uptime information is not available to virtual host administrators.

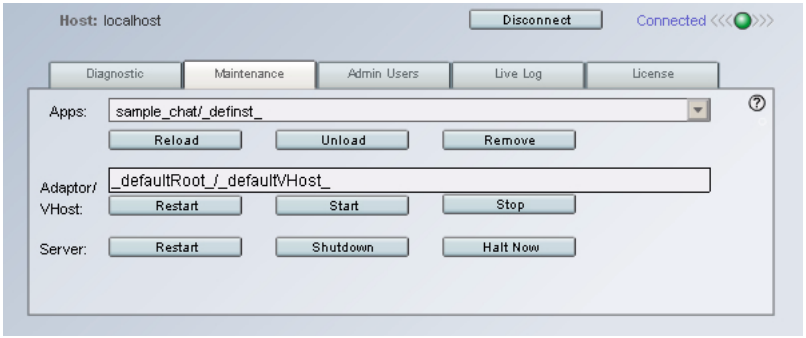
Ping lets you verify that the server is running and view its responsiveness in milliseconds.

Update allows you to get current information about the application instances that are running on the server. The name of each application is displayed, along with the number of instances of the application that have been created on or unloaded from the server, the number of users that are connected, and the total number of connections that have been accepted and rejected for each application. If you are connected as a virtual host administrator, Update displays information only for the virtual host you are connected to.

An application may have more than one instance if its clients connect to the server by using different application instance parameters within their ActionScript NetConnection calls. (For more information on the NetConnection object, see the *Client-Side Communication ActionScript Dictionary*.)

Using the Maintenance panel

To restart or stop the server, a virtual host, or a client application or application instance, you use the Maintenance panel.



Virtual host administrators can shut down only the applications on their own virtual host and restart only that virtual host.

The Maintenance panel includes the following items:

Apps allows you to choose from the applications currently running on the server. The Apps menu will not update while it is open or selected; to allow it to update, click outside the menu. To specify an application instance that is not listed, type the name of the instance into the text box.

Reload lets you reload an application instance that is currently connected to the server. You might do this to reload the instance's server-side scripts or to disconnect all of its users while immediately allowing new connections. To reload an application instance, select it from the Apps menu and click Reload. (You cannot reload applications that have been unloaded with the Remove button).

Unload lets you delete an application instance. To unload an application instance, select it from the Apps menu and click Unload. This disconnects all the clients of that instance. (If the application has more than one instance on the server, only the instance you choose is deleted).

Remove lets you delete all elements of an application. This includes the application instances, the application directory and any script files it contains, and any stream files associated with the application.

Adaptor/VHost allows you to enter the name of the virtual host you want to control. If you want to perform functions on the virtual host you are currently connected to, you can leave this text box blank. To specify another virtual host, enter the name of its adaptor and the name of the virtual host, separated by a forward slash (/), for example, _Root1_/www.macromedia.com. The Adaptor/VHost text box is not visible to virtual host administrators because they can only connect to and control their own virtual host. If you are running a developer edition of the server, you do not need to use the Adaptor/VHost text box because only one virtual host is allowed on the server in addition to the default virtual host.

VHost Restart lets you restart a virtual host. Restarting a virtual host disconnects all users of that virtual host and removes all application instances from memory. If you are a server administrator you can restart any of the server's virtual hosts except the default virtual host (_defaultVHost_). To restart a virtual host, enter its name in the Adaptor/VHost text box and click Restart. If you are a virtual host administrator, you can only connect to and administer your own virtual host. Click Restart to restart the virtual host.

Start lets you start a virtual host that has been previously stopped. This button is available only to server administrators. If you are a server administrator, you can start any of the server's virtual hosts except the default virtual host (_defaultVHost_). To start a virtual host, enter its name in the Adaptor/VHost text box and click Start.

As a security feature, virtual host administrators are not allowed to start a virtual host that has been stopped.

Stop lets you stop a virtual host. This button is available only to server administrators. Stopping a virtual host disconnects all users from that virtual host and prevents future connections. If you are a server administrator you can stop any of the server's virtual hosts except the default virtual host (_defaultVHost_). To stop a virtual host, enter its name in the Adaptor/VHost text box and click Stop.

Server Restart lets you restart the server. This button is available only to server administrators. Restarting the server disconnects all connected users and unloads all application instances on all virtual hosts from memory. A restart also reloads the server's configuration files, allowing any manually entered changes to those files to be read by the server.

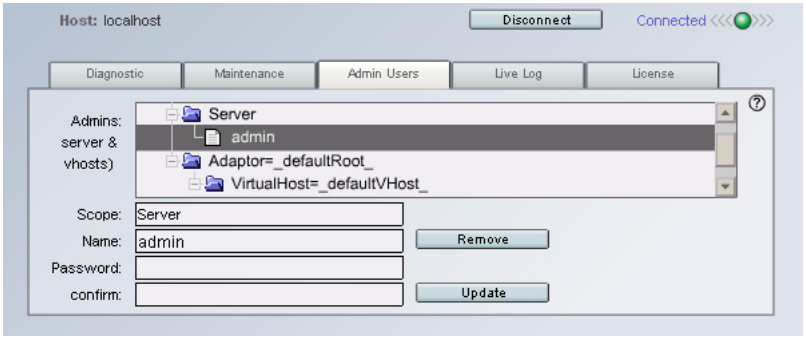
Shutdown lets you exit from the server. This button is only available to server administrators. When you use the Shutdown button, the server unloads all application instances and cleans up all streams and other shared objects that the applications may be using. Shutting down can take a few seconds while the server performs this cleanup procedure. When the server is shut down, the Administration Console is disconnected.

Halt Now allows you to force the server to shut down immediately without any cleanup of streams or shared objects. This button is only available to server administrators. Because the server terminates without cleanup, data from all applications in memory will be lost. Therefore, use Halt Now only when other methods of unloading applications, stopping virtual hosts, or stopping the server fail.

Messages appear at the bottom of the Maintenance panel to indicate the success or failure of each operation you perform.

Adding and editing administrators

To add or edit administrator log-on information, you use the Admin Users panel.



The Admin Users panel contains the following items:

Admins displays a hierarchical view of the server's administrators, adaptors, virtual hosts, and virtual host administrators. Virtual host administrators can only see the administrators of their own virtual host. To perform an operation in the Admin Users panel, you first select a server, virtual host, or administrator icon in the Admins pane; then you use the text boxes and buttons to finish the operation.

Scope displays the level of server access of the currently selected administrator. If the selected administrator is a server administrator, "Server" appears in the Scope text box. If the selected administrator is a virtual host administrator, that administrator's adaptor and virtual host appear in the Scope text box.

Name displays the user name of the currently selected administrator. When adding new administrators, you enter the user name in the Name text box.

Password allows you to enter a password when adding a new administrator or editing an existing one. Administrators cannot have empty passwords.

Confirm allows you to reenter the password when adding or updating an administrator. The password must match the one entered in the Password text box for the operation to be successful.

Update/Add allows you to add a new administrator or change an existing administrator's password; the button is labeled Add when the server or a virtual host is selected in the Admins pane. The button is labeled Update when an existing administrator is selected in the Admins pane. Virtual host administrators can update their own password information on this panel. Only server administrators can add or change the passwords of other virtual host administrators.

Remove allows you to delete the currently selected administrator. Only server administrators can use this function.

To add an administrator:

1 Select the server or the virtual host you want to add the administrator to.

The server or virtual host name appears in the Scope text box.

2 Enter the new administrator's user name in the Name text box.

3 Enter the new administrator's password in the Password text box.

4 Reenter the password in the Confirm text box.

5 Click the Add button.

To update an administrator's password:

1 Select the administrator in the Admins pane.

2 Enter the new password in the Password text box.

3 Reenter the password in the Confirm text box.

4 Click Update.

To remove an administrator:

1 Select the administrator in the Admins pane.

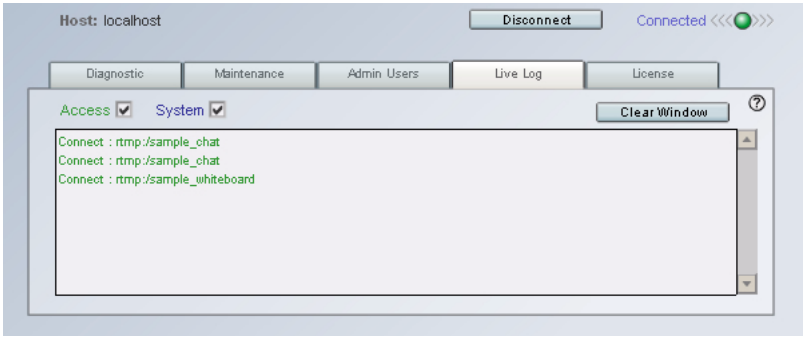
2 Click Remove. Be sure not to remove the last server administrator.

Note: When you use the Admin Users panel, the Server.xml file must be writable and not open in any other application.

You can also add and edit administrator information by manually editing the server's Server.xml file. Changes made in this way will not take effect on the server until after a server restart. For more information about editing the server's configuration files, see About the configuration files.

Viewing server logs

To view a list of server activity, you use the Live Log panel. The Live Log panel is not available to virtual host administrators because it may contain sensitive information.



The Live Log panel includes the following options:

Access lets you view information about user connections and disconnections to and from the server. When you select this option, these items are displayed in green in the output pane.

System lets you view server error messages, such as attempts to connect to nonexistent applications. When you select this option, server messages are displayed in blue.

Clear Window lets you clear the log text from the pane.

Viewing and editing license information

To view or change the server's license key and its capacity parameters, you use the License panel. This panel is only available to server administrators.

To change the license key, enter the new key and click Add. Entering a new license key will write the new key to the Server.xml file, but will not change the key that is in memory on the server. To assert the new key on the server, you must restart the server in the Maintenance panel (see Using the Maintenance panel).

When updating the license key, be sure the Server.xml file is writable and not open in any other application.

Performing advanced administration tasks in Windows

Once you're familiar with basic Flash Communication Server administration tasks, you can make operating and troubleshooting the server more convenient and effective.

Advanced administration tasks include the following:

- Viewing server events in the Windows event viewer
- Configuring the server at runtime

Viewing server events in the Windows event viewer

In addition to the Communication App inspector and NetConnection Debugger in Flash MX, you can also use the Windows event viewer for tracking Flash Communication Server activity and debugging server applications. The event viewer displays a list of events that the server generates.

To use the Windows event viewer:

1. From the Windows Start menu, select Settings > Control Panel > Administrative Tools > Event Viewer.
2. Select the Application panel.
3. Double-click an event generated by Flash Communication Server to view the details of the event.

Configuring the server at runtime

Using the Server Management ActionScript API, you can view and edit the server's configuration settings by building your own customized administration applications. You can add or remove administrators, change their user names and passwords, and change most of the other server settings in all four of the server's XML files. For detailed information about using these ActionScript commands, see the Flash MX Support Center (www.macromedia.com/support/flash/).

Managing Flash Communication Server on UNIX

On all supported UNIX platforms, Flash Communication Server is installed as a service and includes a command-line utility, the fcsmgr utility, to perform certain administration tasks

You must be a root user to install the server and manage it using the fcsmgr utility; for more information, see Using the fcsmgr utility.

The default ports for the server and the Admin service are, respectively, 1935 and 1111. The default option is to run the server as non-root user 'nobody.' If you don't want to run the server as user 'nobody,' choose a user with a valid account on the system. To specify which user to run the server as after installation, edit the `<Process>` tags in the Server.xml configuration file. Be sure the user you specify has permission to read/write the server files. For more information, see The Server.xml file.

Starting the Admin service

The Admin service is the service that you connect to when you log on to the server through the Administration Console. It is installed automatically when you install the server.

The Admin service must be running to allow users to access the Administration Console. You must be a root user to start the Admin service. To start the Admin service, see Using the fcsmgr utility.

Starting the Flash Communication Server service

The Flash Communication Server service is configured by default to start when the host system is started. You can change this setting using the fcsmgr utility.

You cannot configure an Admin service to start automatically; the Admin service must be started manually by a root user.

Using the fcsmgr utility

Use the fcsmgr utility to perform basic management tasks for the Flash Communication Server service and the Admin service. You must be a root user to use the fcsmgr utility. The following table describes the commands for the fcsmgr utility.

Command	Description
`fcsmgr setautostart`	Sets the Flash Communication Server service to start automatically when the system is started.
`fcsmgr clearautostart`	Sets the Flash Communication Server service to be started manually. This command only affects the server service; Admin services cannot be started automatically.
`fcsmgr server start`	Starts the Flash Communication Server service.

Command	Description
`fcsmgr server stop`	Stops the Flash Communication Server service.
`fcsmgr server restart`	Stops a running Flash Communication Server service and restarts it.
`fcsmgr server abort`	Stops a running Flash Communication Server. Processes, such as streaming or garbage collection, are not allowed to complete.
`fcsmgr adminserver start|stop|abort|restart`	Starts, stops, restarts, or aborts the Admin service.

For other administrative tasks, such as adding users or checking the status of applications, you use the Administration Console. For more information, see Using the Administration Console. Although you do not need to be a root user to use the Administration Console, the Admin service itself does need to be started by a root user using the fcsmgr utility before anyone can use the Administration Console.

CHAPTER 3
Configuring Flash Communication Server

Macromedia Flash Communication Server MX 1.5 has been designed to accommodate many types of communication applications. After installation, the server's configuration files contain only simple, generic settings. You'll need to make some decisions about how to configure it to best suit your needs.

This chapter describes the server's initial configuration and the XML files that define the configuration. It then explains how to edit these files to suit the needs of the client applications you intend to use.

Typical configurations

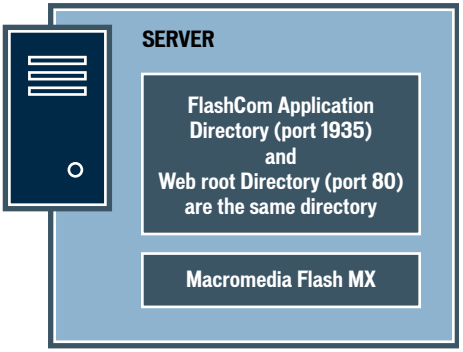
Flash Communication Server can be used in a variety of different ways, with different configurations. In most cases, the server will be used in conjunction with a web server. As described in Chapter 2, "Managing the Server," on page 11, applications that run on Flash Communication Server consist of clients developed in Macromedia Flash MX (SWF files); directories on the server register the application on the server and contain streams and scripts used by the application.

Your web server is responsible for serving the SWF client files and the HTML pages in which they are embedded. In addition, you may use an application server in conjunction with your web server and Flash Communication Server to incorporate database or other features into your communication applications.

If you are running Flash Communication Server on a UNIX system, you must also have a Windows or Macintosh computer with Flash MX to create Flash Communication Server applications. Your client-side files (SWF and HTML files) for your applications can reside on the Flash Communication Server computer or another computer, depending on where your web server is installed. Your source files and server-side scripts (ASC, FLA, FSO, and FLV files) must reside on the UNIX computer where Flash Communication Server is running.

The following diagrams illustrate some typical deployment scenarios.

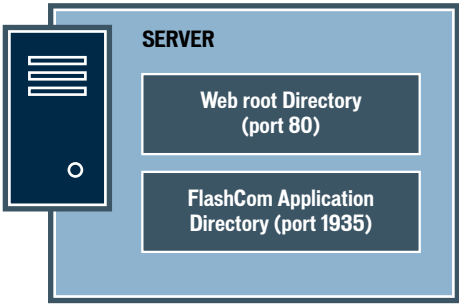
Development During the development process, you may choose to use one computer with a web server, Flash Communication Server, and Flash MX installed.


SERVER
FlashCom Application Directory (port 1935) and Web root Directory (port 80) are the same directory
Macromedia Flash MX

The web root directory in this scenario would contain all the components of your applications, such as the application's FLA, SWF, HTML, script, stream and shared object files. This setup provides a simple working environment for designing and testing your applications.

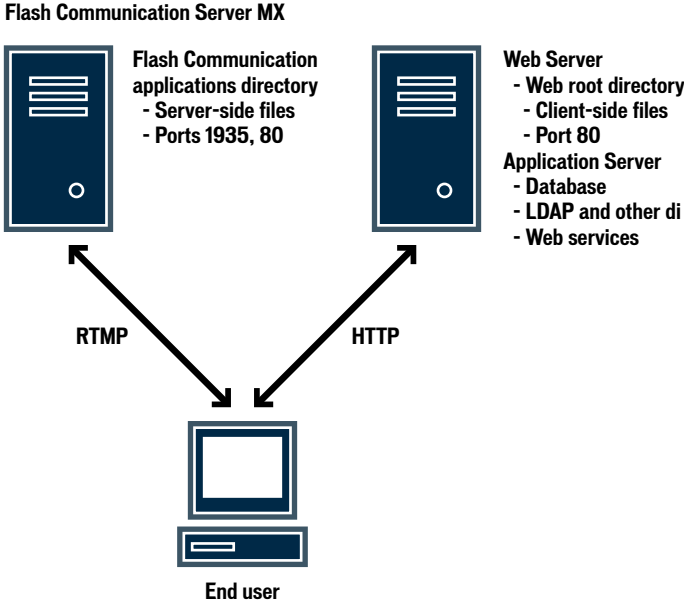
For security reasons, this configuration is not recommended for deployment. In deployment, your FLA, script, stream, and shared object files would not be stored in the web root directory. This configuration is possible only if you have installed Flash Communication Server for Windows.

Deployment: one computer A relatively simple deployment scenario consists of one computer with a web server, Flash Communication Server MX, and a firewall installed. The firewall provides security for the server computer and the rest of your local network.



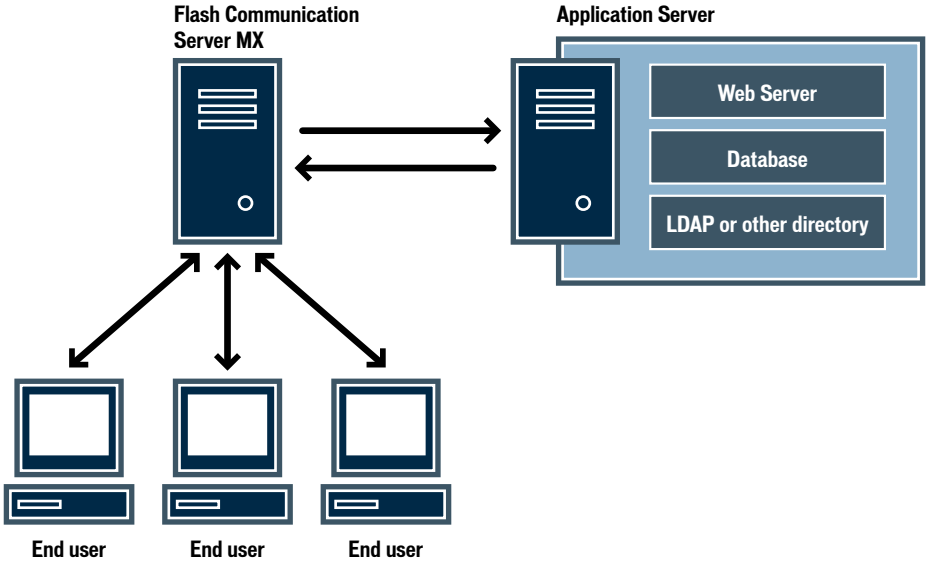
In this and any other deployment scenario, the server-side files (ASCs), the audio/video files (FLVs), and the source files (FLAs) should not reside in the web server's published directories. These files should be located in your registered application directory within the applications directory in the Flash Communication Server directory (or, if you changed this location, the directory in the `<AppsDir>` tag in the Vhost.xml file). The web server's web root directory should contain only the HTML and SWF files for your applications.

Deployment: two computers In this scenario the Flash Communication Server and application server are on two separate computers. This allows for more separation of files and functions and more processor bandwidth. The web server computer can also host an application server if your situation requires one.

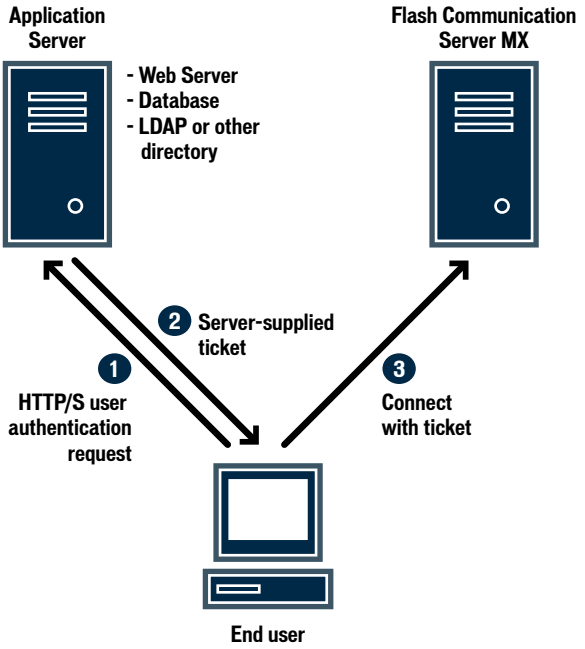


If you are running Flash Communication Server on a UNIX system, you might also use this configuration for development purposes, in which case you would also install Flash MX for Windows or Macintosh on the web server computer.

Deployment: two computers with authentication via Flash Communication Server Some scenarios may require authentication of users who want to access information on an application server. In this case you may want to use a separate computer for Flash Communication Server MX, and another for the web server and application server. Your Flash Communication Server can perform the authentication and then retrieve data from the web/application server. This scenario requires the developer to create server-side scripts to perform these functions.



Deployment: two computers with authentication via an application server You may decide to have users authenticated before they are allowed to connect to your Flash Communication Server. In this scenario, users connect first to the web/application server. If they are authenticated, the application server creates a ticket that the user then uses to connect to the Flash Communication Server. The Flash MX communication application must be designed to check for these tickets, typically with server-side scripts.



About configuration levels

The server is capable of hosting more than one adaptor and more than one virtual host on each adaptor. Each virtual host is equivalent to a domain name. Each virtual host can run more than one communication application. (Some editions of the server are limited to one adaptor and one virtual host.)

When you're ready to customize the server for your own virtual hosts and applications, you'll edit the server's XML configuration files and the directory structure that contains them. The server uses four configuration files in XML format: Server.xml, Adaptor.xml, Vhost.xml, and Application.xml. You configure the server by editing the contents of these files, either in a text editor or with the Administration Console.

The following sections explain in detail how to configure the server using these files.

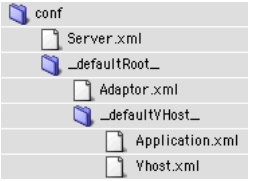
Note: If you edit any XML configuration files or create any new ones, you must restart the server before the new settings take effect.

About the configuration hierarchy

Flash Communication Server can support several adaptors, virtual hosts, and applications simultaneously. Each adaptor on the server can serve multiple virtual hosts, and each virtual host can host multiple applications. By supporting multiple adaptors and virtual hosts, Flash Communication Server facilitates management of multiple websites that may have different configurations and administrators.

Each of these layers of service has its own configuration settings, stored in separate XML files. These files are stored in a directory structure that reflects the hierarchy of adaptors, virtual hosts, and applications you want to use with the server.

The default directory structure installed with the server looks like this:



The directory structure includes three subdirectories: conf, _defaultRoot_, and _defaultVHost_.

- The conf subdirectory, at the top of the hierarchy, holds all the configuration files for the server. This subdirectory contains the Server.xml file and the _defaultRoot_ subdirectory. The Server.xml file contains settings that relate to the server only. The specific settings for the adaptors, virtual hosts, and applications are stored in separate XML files.
- The _defaultRoot_ subdirectory is the default adaptor directory for the server. It contains the Adaptor.xml file and the _defaultVHost_ subdirectory. The Adaptor.xml file contains the settings that relate to the adaptor. (If there were a separate adaptor, it would have its own subdirectory at the same level as the _defaultRoot_ subdirectory.)
- The _defaultVHost_ subdirectory is the default virtual host subdirectory for the adaptor. It contains the Vhost.xml file, which contains the settings for the virtual host, and the Application.xml file, which contains default settings for the client applications that will connect to the server. (If there were another virtual host on the same adaptor, it would have its own subdirectory at the same level as the _defaultVHost_ subdirectory.) Each adaptor directory must contain a _defaultVHost_ directory.

Adding adaptors and virtual hosts

To add an adaptor to the server, you must add a complete adaptor directory structure to the server's conf directory. Each adaptor directory must contain an Adaptor.xml file and at least one virtual host directory, called _defaultVHost_. When you add virtual hosts, these must be in addition to _defaultVHost_.

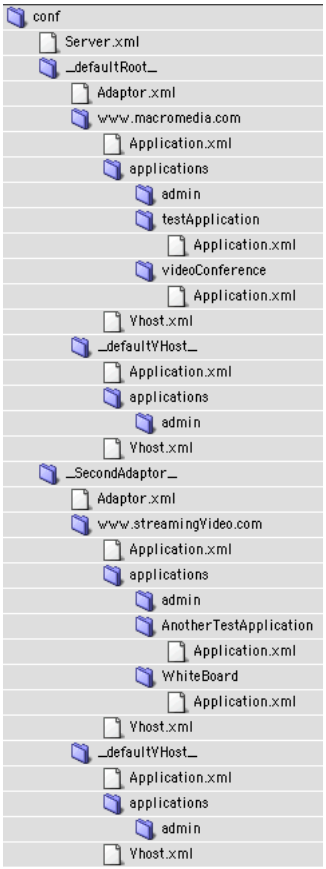
When you design an application in Macromedia Flash MX that will connect to the Flash Communication Server, you add a subdirectory with the application's name to the application directory of the virtual host it will connect to. The application's subdirectory can contain its own Application.xml file if you want the application to override any of the settings in the generic Application.xml file in the virtual host directory.

To create a new virtual host, create a new virtual host directory inside the /conf/*adaptor_name* directory in the Flash Communication Server directory, for the adaptor you wish to use for the new virtual host:
/conf/*adaptor_name*/*virtual_host_name*. This directory must include the following items:

- A Vhost.xml file
- An Application.xml file

You also need to add the new virtual host to the Server.xml file and add administrative users for that virtual host.

A typical customized server conf directory might look like this:



A customized conf directory containing multiple adaptor, virtual host, and application subdirectories and configuration files

The conf directory illustrated here contains two adaptor subdirectories: the _defaultRoot_ subdirectory and the _SecondAdaptor_ subdirectory.

- The _defaultRoot_ subdirectory contains the _defaultVHost_ subdirectory and another virtual host subdirectory named www.macromedia.com. Each of these virtual host subdirectories contains an application subdirectory. The application subdirectory for www.macromedia.com contains directories for the applications testApplication and videoConference.
- The _SecondAdaptor_ subdirectory contains a _defaultVHost_ directory and another virtual host directory named www.streamingVideo.com. The www.streamingVideo.com subdirectory contains an application subdirectory with directories for AnotherTestApplication and a WhiteBoard application.

Tip: The /conf/adaptor_name/vhost_name/applications directory is different from the applications directory you use to register your applications.

About the configuration files

Flash Communication Server uses separate XML configuration files for each level of the server hierarchy: server, adaptor, virtual host, and application. Each of these files contains configuration tags that relate to the server, adaptor, virtual host, or application they are associated with. To customize the functionality of the server, you edit these tags. The server edits some of these tags itself when you use the Administration Console.

If you edit any XML configuration files, you must restart the server before the new settings take effect.

The following sections describe the tags in each XML file in detail.

The Server.xml file

The Server.xml file is located at the root level of the conf directory and contains tags that define user names and passwords for server administrators and virtual host administrators. You edit the Server.xml file to add or remove administrators. The administrators defined in this file will be able to connect to the server with the Administration Console. Server administrators can use all of the features and see all of the information available in the Administration Console. Virtual host administrators can use the Administration Console only to see information and perform tasks that relate to their particular virtual host.

The Server.xml file contains the following tag structure, along with comments that briefly describe each tag.

```
<Root>
  <Admin>
    <Server>
      <UserList>
        <User name="acme">
          <Password encrypt="true">YWRtaW4=</Password>
          <Allow></Allow>
          <Deny></Deny>
          <Order>Deny,Allow</Order>
        </User>
      </UserList>
    </Server>
    <Adaptor name= "_defaultRoot_">
      <VirtualHost name= "_defaultVHost_">
        <UserList>
          <User name= "">
            <Password encrypt "false"></Password>
            <Allow></Allow>
            <Deny></Deny>
            <Order>Deny,Allow</Order>
          </User>
        </UserList>
      </VirtualHost>
    </Adaptor>
  </Admin>
  <Server>
    <Process>
      <UID></UID>
      <GID></GID>
    </Process>
    <Mask>017</Mask>
    <LicenseInfo>FSW100-XXXXX-XXXXX-XXXXX-XXXXX</LicenseInfo>
    <AdminServer>
      <HostPort>:1111</HostPort>
      <SocketGC>60</SocketGC>
      <Process>
        <UID></UID>
        <GID></GID>
      </Process>
    </AdminServer>
    <ResourceLimits>
      <CPUMonitor>1</CPUMonitor>
      <ThreadPoolGC>20</ThreadPoolGC>
      <ApplicationGC>5</ApplicationGC>
      <FLVCacheSize>40</FLVCacheSize>
    </ResourceLimits>
    <ServerDomain></ServerDomain>
  </Server>
</Root>
```

The tags in the Server.xml file serve the following purposes:

- `<Root>` is the root tag for the XML document. It is a container for all the other tags.
- `<Admin>` is a container for server administrator and virtual host administrator settings.
- The first `<Server>` tag is a container for the server administrator `<UserList>` tags.
- `<UserList>` is a container for one or more `<User>` tag groups.

- The <User> tag inside the <UserList> tag specifies the user name of a server administrator in its name parameter and is a container for the <Password>, <Allow>, <Deny>, and <Order> tags.

  To connect with the Administration Console that is installed with the server, a user must be defined in a <User> tag group. For more information about using the Administration Console (admin.swf), see Using the Administration Console.

  You define the first administrator when you run the server installer. There must be at least one administrator defined for the server; otherwise no users can connect with the Administration Console. You can define additional administrators by including additional <User> sections in the Server.xml file. Each <User> section must contain a <Password> tag. Normally, you edit administrator user names and passwords using the Administration Console. You may use any characters in user names entered with the Administration Console. However, if you edit this information directly in the Server.xml file, remember to use only valid XML.

  By default, only the Administration Console requires a valid administrator user name and password when a connection is being made to the server. Other client applications do not require any user name or password unless they are explicitly programmed to do so by the author.

- <Password> specifies the password for the <User> tag that contains it. The encrypt attribute indicates whether to encrypt the contents of the password. If the encrypt attribute is set to true, the password you see in the file is the encrypted password. If you edit the contents of this tag directly in the Server.xml file rather than with the Administration Console, you must set the encrypt attribute to false and use only valid XML in the password. Passwords cannot be empty ("").

- <Allow> contains a list of hosts from which the administrator user should be allowed to connect. You can include whole host names (also called domain names) or IP addresses in the list. Separate each host name or address with a comma. The keyword all can be specified to allow connections from all hosts.

  For example, an <Allow> tag might look like this:

  ```
  <Allow>www.macromedia.com, 12.34.56.78</Allow>
  ```

  Whenever possible, use IP addresses in the <Allow> tag. This increases the server's performance when processing connection requests.

- <Deny> contains a list of hosts from which the administrator user should not be allowed to connect. You can include whole or partial host names (also called domain names) or IP addresses in the list. By including partial host names, such as myCompany.com, you can deny connections to users connecting from any computer within the myCompany.com domain. Separate host names or addresses with a comma. The keyword all can be specified to deny connections from all hosts except the local host.

  For example, a <Deny> tag might look like this:

  ```
  <Deny>hackerSite.com, 87.65.43.21</Deny>
  ```

  As with the <Allow> tag, use IP addresses in the <Deny> tag whenever possible for increased server performance.

- <Order> specifies the order in which to evaluate the preceding <Allow> and <Deny> tags. It can be set to either Deny,Allow (the default) or Allow,Deny.

  When Deny,Allow is specified, the server allows the user specified in <Username> to connect from any host that is not included in the <Deny> tag or is included in the <Allow> tag. Access is denied from any host that is included in the <Deny> tag and is not included in the <Allow> tag.

  When Allow,Deny is specified, the server allows the specified user to connect only from hosts that are included in the <Allow> tag and are not included in the <Deny> tag. Access is denied from any host that is not included in the <Allow> tag or is included in the <Deny> tag.

- <Adaptor> specifies the name of an adaptor in its name attribute. It is a container for one or more <VirtualHost> tag groups that specify the virtual host administrators for each virtual host in the adaptor. You may specify multiple adaptors by adding additional <Adaptor> tag groups to the Server.xml file.

- <VirtualHost> specifies the name of a virtual host in its name parameter. It is a container for a <UserList> tag that defines administrators for the specified virtual host. The name parameter specifies the name of the virtual host; the name of the virtual host should be the same as used in the NetConnection.connect statement of an application that connects to this host.

  The name parameter also tells the server which virtual host the enclosed <UserList> tag refers to. If you have more than one virtual host on one or more adaptors on the server, you specify their administrator information with separate sets of <VirtualHost> tags in the Server.xml file.

- The <UserList> tag is a container for one or more <User> tag groups.

- The next <User> tag specifies the name of a virtual host administrator in its name parameter. The user name can contain only alphanumeric characters. The <User> tag contains the same set of password and host-permission tags as the <User> tag in the earlier <Server> tag, but this <User> tag specifies the administrator user information for the virtual host tag that contains it. Virtual host administrators can only perform administration tasks relating to the applications running on their own virtual host. Administration tasks relating to the virtual host itself must be performed by a server administrator.

  You can specify multiple administrator users for a single virtual host by including additional sets of <User> tags in the <UserList> tag for the virtual host.

  Only users defined here and in the earlier <User> tag (in the <Server> section) are allowed to connect to the Administration Console. (For more information, see Using the Administration Console.) Normally, you add and edit virtual host administrator information with the Administration Console. If you edit user names directly in the Server.xml file, you must use only valid XML in the <User> and <Password> tags.

  By default, only the Administration Console requires a valid administrator user name and password when connecting to the server. Other client applications do not require any user name or password unless they are explicitly programmed to do so by the author.

- The <Password> tag specifies the password for the <User> tag that contains it. Its format is the same as the <Password> tag inside the earlier <Server> tag.

- The <Allow> tag inside the <VirtualHost> tag works the same way as the <Allow> tag inside the earlier <Server> tag, but applies only to the <User> tag that contains it.

- The <Deny> tag inside the <VirtualHost> tag works the same way as the <Deny> tag inside the earlier <Server> tag, but applies only to the <User> tag that contains it.
- The <Order> tag inside the <VirtualHost> tag works the same way as the <Order> tag inside the earlier <Server> tag, but applies only to the <User> tag that contains it.
- The <Server> tag contains the <Process> tag, the <Mask> tag, the <LicenseInfo> tag, the <AdminServer> tag, and the <ResourceLimits> tag. This <Server> tag is distinct from the <Server> tag found under the <Admin> tag earlier in the file.
- <Process> contains the next two tags, <UID> and <GID>. These tags are applicable for Flash Communication Server systems running on UNIX systems only. At installation, you specify which user (the default is the "nobody" user) server processes should run as. This user is also the owner of the Flash Communication Server files. To make server processes run as a different user than what you specified at installation, you can use the <UID> and <GID> tags. Note that if <UID> and <GID> are empty, the server runs as root.
- <UID> contains the process user ID (valid for UNIX systems only). The default value is 99 for the "nobody" user. If this or the <GID> tag are empty, the server runs as root.
- <GID> contains the process group ID (valid for UNIX systems only). The default value is 99 for the "nobody" group. If this or the <UID> tag are empty, the server runs as root.
- <Mask> contains a three-digit octal value used by the umask (user permissions mask) command to set a file creation mask. It controls who has read/write access to shared object and stream files in the server. This tag is applicable for UNIX systems only. The default mask value is 017; the server attempts to create files with a permissions setting of 666 and applies 666 to the 017 mask, resulting in a permissions setting of 0660. Acceptable values for each digit are from 0 to 7; if an invalid value is given, the default mask is applied.

  The default value indicates that the owner of the server files and all the users belonging to the same group as the owner will have read/write access to shared object and stream files. (The owner is the root user or the user as which the server processes run, as specified during installation or in the <UID> tag.)

  Note: Regardless of the value of <Mask>, the server will never create files with execute permissions. Also, be careful if you change the first digit of <Mask>, which controls permissions for the owner: if the first digit is set to anything other than 0 or 1, the owner is denied read/write access, and the server itself will not be able to access the files it creates. All users will be denied the ability to record streams, keep persistent shared objects, or access FLV files or other features related to shared objects. For more information on umask usage, consult a standard UNIX reference manual.
- <LicenseInfo> contains the serial number for the server. The value of this tag is written to the Server.xml file by the server installer during installation.
- <AdminServer> is a container for the <HostPort> tag, the <SocketGC> tag, and the <Process> tag.
- <HostPort> specifies the port that the Admin service binds to. The Admin service is separate from the Flash Communication Server service. When administrators connect to the server with the Administration Console, they are connecting to the Flash Communication Admin service, which in turn connects to the Flash Communication Server. The default value is :1111 (note the colon). Only one port number may be specified in this tag.
- <SocketGC> specifies how often, in seconds, to check for and remove inactive sockets on the Admin service.

- <Process> contains the next two tags, <UID> and <GID>. These tags are applicable for Flash Communication Server systems running on UNIX systems only. At installation, Admin service processes are defined to run as a root user. You can specify a user ID and group ID for Admin service processes using these tags.
- <UID> contains the process user ID. The default value is "nobody". If this and the <GID> tag are empty, the Admin service runs as root.
- <GID> contains the process group ID. The default value is "nobody". If this and the <UID> tag are empty, the Admin service runs as root.
- <ResourceLimits> specifies the maximum resource limits for the server. It contains the next five tags.
- <CPUMonitor> specifies, in seconds, how often to monitor CPU usage. The default value is 1 second.
- <ThreadPoolGC> specifies, in minutes, how often to collect and dispose of unused I/O threads. The default value is 20 minutes.
- <ApplicationGC> specifies, in minutes, how often to collect and dispose of unused application instances. The default and minimum value is 5 minutes.
- <FLVCacheSize> specifies the maximum FLV cache size, as a percentage of the total physical memory on the system. The default value is 40%.
- <ServerDomain> specifies the domain name of the server. Set this to the server's domain name so that it can pass the domain name to any application servers it connects to. For security purposes, some application servers require this information as a part of incoming connection requests.

The following set of Server.xml tags has been customized for real-world use. Two server administrators are defined, `root` and `jsmith`. An administrator named `panderson` is defined for two virtual hosts, _defaultVHost_ and mysite.com. The `<Allow>` and `<Deny>` tags are set for all four administrators to allow them to connect only from the IP address 12.34.56.78. The server domain is set to www.macromedia.com.

```
<Root>
  <Admin>
    <Server>
      <UserList>
        <User name="root">
          <Password encrypt="true">WTs5Ka9</Password>
          <Allow>12.34.56.78</Allow>
          <Deny>all</Deny>
          <Order>Allow,Deny</Order>
        </User>
        <User name= "jsmith">
          <Password encrypt="true">4flYnH1</Password>
          <Allow>12.34.56.78</Allow>
          <Deny>all</Deny>
          <Order>Allow,Deny</Order>
        </User>
      </UserList>
    </Server>
    <Adaptor name="_defaultRoot_">
      <VirtualHost name="_defaultVHost_">
        <UserList>
          <User name="panderson">
            <Password encrypt="true">jhdAYT2E7</Password>
            <Allow>12.34.56.78</Allow>
            <Deny>all</Deny>
            <Order>Allow,Deny</Order>
          </User>
        </UserList>
      </VirtualHost>
      <VirtualHost name="mysite.com">
        <UserList>
          <User name="panderson">
            <Password encrypt="true">jhdAYT2E7</Password>
            <Allow>12.34.56.78</Allow>
            <Deny>all</Deny>
            <Order>Allow,Deny</Order>
          </User>
        </UserList>
      </VirtualHost>
    </Adaptor>
  </Admin>
  <Server>
    <Process>
      <UID></UID>
      <GID></GID>
    </Process>
    <Mask>017</Mask>
    <LicenseInfo>FSW100-12345-67890-54321</LicenseInfo>
    <AdminServer>
      <HostPort>:1111</HostPort>
      <SocketGC>60</SocketGC>
      <Process>
        <UID></UID>
        <GID></GID>
      </Process>
```

```
    </AdminServer>
    <ResourceLimits>
      <CPUMonitor>1</CPUMonitor>
      <ThreadPoolGC>20</ThreadPoolGC>
      <ApplicationGC>5</ApplicationGC>
      <FLVCacheSize>40</FLVCacheSize>
    </ResourceLimits>
    <ServerDomain>www.macromedia.com</ServerDomain>
  </Server>
</Root>
```

The Adaptor.xml file

The Adaptor.xml file defines settings for an adaptor. It determines the number of threads that can be used by the adaptor, the communications ports that adaptor binds to, and the IP addresses or domains from which the adaptor can accept connections.

Each adaptor must have its own directory inside the server's conf directory. The name of the directory is the name of the adaptor. Each adaptor directory must contain an Adaptor.xml file. For example, the default adaptor included with the server at installation is named _defaultRoot_, and its directory is found in the \Flash Communication Server MX\conf\ directory.

To change an adaptor's settings, you edit the tags in the Adaptor.xml file.

The Adaptor.xml file contains the following tag structure, along with brief comments for each tag:

```
<Adaptor>
  <ResourceLimits>
    <MaxThreads>100</MaxThreads>
  </ResourceLimits>
  <HostPortList>
    <HostPort>:1935</HostPort>
  </HostPortList>
  <Allow></Allow>
  <Deny></Deny>
  <Order>Deny,Allow</Order>
</Adaptor>
```

The tags in the Adaptor.xml file serve the following purposes:

- `<Adaptor>` contains all the other adaptor configuration tags.
- `<ResourceLimits>` is a container for the `<MaxThreads>` tag.
- `<MaxThreads>` specifies the maximum number of threads to create on the server for processing input/output requests. The default is 100. Increasing this number allows more threads to run concurrently, but also increases the processor workload by requiring more thread-switching operations. Generally you will not need to change this setting. If you do want to change it, test you applications thoroughly to determine the best value.
- `<HostPortList>` is a container for one or more `<HostPort>` tags.

- `<HostPort>` specifies the IP address and one or more port numbers for the adaptor to bind to. The IP address and port number are separated by a colon. If you specify more than one port, separate them with commas.

  For example, a `<HostPort>` tag might look like this:

  ```
  <HostPort>12.34.56.78:1935, 443</HostPort>
  ```

  You can set up the adaptor to listen on more than one IP address by specifying more than one `<HostPort>` tag.

  By default, Flash Communication Server uses port number 1935 for RTMP connections. You may choose to use other ports. When there is a proxy server or firewall that allows only HTTP content through, the server uses HTTP to transmit RTMP packets (this process is called HTTP tunneling); if a connection through port 1935 fails, the server tries again using HTTP and port 80. To explicitly use HTTP tunneling, you can configure the adaptor to listen on port 80 in the `<HostPort>` tag. Whichever port you use, be sure it is set to the "open" state. When you change the port number, you must also change it in the client application's ActionScript `NetConnection` call.

  As a security precaution, when you specify an IP address in the `<HostPort>` tag, the server will not bind to (listen to) localhost. (The term *localhost* refers to the computer the server is running on.) If you do not specify an address, the server will bind to both localhost and the true IP address of the computer it is running on.

- `<Allow>`, `<Deny>`, and `<Order>` serve the same purposes as in the Server.xml file, but indicate permissions specifically for this adaptor.

The following set of Adaptor.xml tags has been customized for real-world use. Up to five threads can be created for the adaptor. The adaptor will bind to ports 1935 and 80 and will accept connections from any IP address.

```
<Adaptor>
  <ResourceLimits>
    <MaxThreads>5</MaxThreads>
  </ResourceLimits>
  <HostPortList>
    <HostPort>12.34.56.78:1935, 80</HostPort>
  </HostPortList>
  <Allow>all</Allow>
  <Deny></Deny>
  <Order>Deny,Allow</Order>
</Adaptor>
```

The Vhost.xml file

The Vhost.xml file configures a virtual host within an adaptor. Each virtual host must have its own directory inside the adaptor directory.

The name of the directory must be the actual name of the virtual host, such as streaming.macromedia.com. Each virtual host you define must map to a DNS entry, or another name resolution such as a WINS address or a hosts file, that specifies an IP address on the server computer.

Each adaptor must contain a _defaultVHost_ directory in addition to the custom virtual hosts that you define. If a client application tries to connect to a virtual host that does not exist, the server attempts to connect it to _defaultVHost_.

Each virtual host directory contains a Vhost.xml file. This file contains tags that define the settings for the virtual host. These settings include aliases for the virtual host, the location of the virtual host's application directory, limits on the resources the virtual host can use and other parameters.

The Vhost.xml file contains the following tag structure, with brief comments for each tag:

```
<VirtualHost>
  <AliasList>
    <Alias name="acme">acme.myDomain.com</Alias>
  </AliasList>
  <AppsDir>C:\Program Files\Macromedia\Flash Communication Server
  MX\applications\</AppsDir>
  <RecordAccessLog>true</RecordAccessLog>
  <ResourceLimits>
    <MaxConnections>-1</MaxConnections>
    <MaxAppInstances>-1</MaxAppInstances>
    <MaxStreams>-1</MaxStreams>
    <MaxSharedObjects>-1</MaxSharedObjects>
    <AppInstanceGC>20</AppInstanceGC>
  </ResourceLimits>
  <VirtualDirectory>
    <Streams></Streams>
  </VirtualDirectory>
  <DNSSuffix></DNSSuffix>
  <Allow>all</Allow>
</VirtualHost>
```

The tags in the Vhost.xml file are described below. You can change the values of some of these tags at runtime; to learn how, see the administration API article at Macromedia's website (www.macromedia.com/go/flashcom_admin).

- `<VirtualHost>` is the root tag for the XML document. It is a container for all the other tags.
- `<AliasList>` is a container for one or more `<Alias>` tags.
- `<Alias>` allows you to specify an alternative short name to use when connecting to the virtual host. You can use this tag to shorten long host names. You can specify an unlimited number of aliases by adding additional `<Alias>` tags. Each alias you specify must be mapped to the correct IP address for the virtual host.

  For example, if the host name is `machineName.company.com`, then you could use the `name` attribute of the alias tag to shorten the host name to `machineName`:

  ```
  <Alias name = "machineName">machineName.company.com</Alias>
  ```

  Do not use the same alias for more than one virtual host. If more than one virtual host on the same adaptor has the same alias defined, then the first match found will be used by the server. This can cause unpredictable results.

- `<AppsDir>` specifies the path to the applications for this virtual host; during installation, the default location is written to the default Vhost.xml file. In Windows, the default `<AppsDir>` location is C:\Program Files\Macromedia\Flash Communication Server\applications. On UNIX, the default location is /opt/macromedia/fcs/applications.

  If nothing is specified in this tag, it defaults to C:\Program Files\Macromedia\Flash Communication Server MX\conf_defaultRoot__defaultVHost_\ in Windows. On UNIX, the default <AppsDir> path is /opt/macromedia/fcs/conf/_defaultRoot_/_defaultVhost_/.

  This tag allows you to change the location of your applications directory and server-side script files, if you wish. For example, if you want to keep your application subdirectories and server-side script files in C:\Financial_Apps, you can specify that path in the `<AppsDir>` tag. You would create subdirectories named with your application names in C:\Financial_Apps to register your applications with the server.

  You can also specify multiple applications directories by separating locations with a semicolon (;). For example, you could specify the following two locations, each of which contained application subdirectories: `<AppsDir>C:\MyApps;D:\NewApps</AppsDir>`.

  If you change the default location of `<AppsDir>`, be sure to include a directory named admin in each directory. This ensures that the Administration Console (admin.swf) will be able to connect to the virtual host. For more information, see Using the Administration Console.

- `<RecordAccessLog>` indicates whether the server should (`true`) or should not (`false`) record connections to the server. This information is written to an access.flv file in the virtual host's admin\Streams\Logs\ directory inside its application directory.

- `<ResourceLimits>` is a container tag for the next five tags. By specifying values for these five tags, you can ensure that the server does not consume excessive resources for the virtual host, and you can prevent denial of service attacks to the server.

- `<MaxConnections>` specifies the maximum number of simultaneous connections allowed on this virtual host. Connections are denied if the specified limit is exceeded. The default of -1 allows an infinite number of connections. If you supply a different value, it must be a positive integer. If you specify a negative or non-numeric value, the default is used. This value can be changed at runtime.

- `<MaxAppInstances>` specifies the maximum number of application instances that can be loaded by this virtual host. (A chat application, for example, might require more than one instance, because each chat room would be represented by a separate instance of the application on the server.) A Flash movie defines which application instance it is connecting to by the parameters it includes with its ActionScript `connect` call.

  The default value of -1 allows an infinite number of application instances. If you specify a different value, it must be a positive integer. This value can be changed at runtime.

- `<MaxStreams>` specifies the maximum number of streams that the virtual host can create. The default value of -1 allows an infinite number of streams. If you supply a different value, it must be a positive integer. This value can be changed at runtime.

- `<MaxSharedObjects>` specifies the maximum number of shared objects that the virtual host can create. (For more information about shared objects, see *Developing Communication Applications*.) The default value of -1 allows an infinite number of streams. If you supply a different value, it must be a positive integer. This value can be changed at runtime.

- `<AppInstanceGC>` specifies, in minutes, how often to collect and dispose of application instance resources: shared objects, streams, and script engines. The default value is 20 minutes; the minimum value is 5 minutes.

  Garbage collection is processor-intensive. To keep memory as free of unused objects as possible, set this value to a shorter interval. To avoid a brief decrease in performance during garbage collection, set it to a longer interval. This value can be changed at runtime.

- `<VirtualDirectory>` is a container for one or more `<Streams>` tags that specify virtual directories for shared stream resources.

- `<Streams>` allows you to specify a virtual directory for storing stream resources used by more than one application. By using a virtual directory, you can specify a relative path that points to a shared directory used by multiple applications. The contents of this tag must be in the form *`virtualDirectory`*`;`*`actualDirectory`*.

  For example, suppose you specify the following:

  ```
  <Streams>common;C:\FlashComServer\myApplications\shared\resources\</
    Streams>
  ```

  In this case, any application that refers to a stream whose path begins with `common\` will access the item in C:\FlashComServer\myApplications\shared\resources\ regardless of the application's own directory structure. Therefore, if the application VideoConference refers to the item common\video\recorded\June5 and the application Collaboration refers to common\video\recorded\June5, they both actually refer to the item C:\FlashComServer\myApplications\shared\resources\video\recorded\June5.

  If the virtual directory you specify does not end with a backslash, one is added by the server.

  You can specify more than one virtual directory mapping by adding multiple `<Streams>` tags.

- `<DNSSuffix>` allows you to specify a primary DNS suffix for this virtual host, such as myCompany.com or myUniversity.edu. If a reverse DNS lookup fails to return the domain as part of the host name, this tag is used as the domain suffix.

- The `<Allow>` tag lets you specify domain names from which client applications should be allowed to connect to this virtual host. The default value is `all`, which allows connections from any domain. If no value is specified, only connections from the domain that is being connected to are allowed. If you list specific domains in this tag, only connections from those domains are allowed. Separate each domain name in the list with a comma.

The following set of Vhost.xml tags has been customized for real-world use. The alias `acme` is defined for the virtual host. The application directory is set to C:\Inetpub\wwwroot\apps\. The maximum number of simultaneous connections is set to 50. The virtual directory for streams is set to \streamTemp. The domain suffix is set to macromedia.com for reverse DNS lookups. Finally, the virtual host will accept connections from any IP address.

```
<VirtualHost>
  <AliasList>
    <Alias name="acme">acme.myDomain.com</Alias>
  </AliasList>
  <AppsDir>C:\Program Files\Macromedia\Flash Communication Server
  MX\applications\</AppsDir>
  <RecordAccessLog>true</RecordAccessLog>
  <ResourceLimits>
    <MaxConnections>50</MaxConnections>
    <MaxAppInstances>-1</MaxAppInstances>
    <MaxStreams>-1</MaxStreams>
    <MaxSharedObjects>-1</MaxSharedObjects>
    <AppInstanceGC>20</AppInstanceGC>
  </ResourceLimits>
  <VirtualDirectory>
    <Streams>streamTemp;C:\sharedstreams</Streams>
  </VirtualDirectory>
  <DNSSuffix>macromedia.com</DNSSuffix>
  <Allow>all</Allow>
</VirtualHost>
```

The Application.xml file

The Application.xml file contains the settings for applications that will run on the server. These settings include, for example, the size of the Server-Side Communication ActionScript runtime engine, the location at which streams and shared objects are stored, and bandwidth limitations.

Each virtual host can contain multiple Application.xml files. The Application.xml file in the virtual host directory configures the default settings for applications within the virtual host. If you want to have different settings for a particular application, create a specific Application.xml file in the application's registered application directory (for example, ../applications/*app_name*) with the settings you want.

Each tag in the virtual host's Application.xml file can include an optional `override` parameter, as in this example:

```
<LoadOnStartup override="no">false</LoadOnStartup>
```

When this parameter is included in a tag and set to `no`, no application-specific Application.xml files can override that tag's setting. If the tag contains subtags, they also cannot be overridden. When you omit the `override` parameter, the tag can be overridden. Only the `<Client>` tag described in the following text includes an `override="no"` parameter by default.

The Application.xml file contains the following tag structure, with brief comments for each tag:

```
<Application>
  <LoadOnStartup>false</LoadOnStartup>
  <MaxAppIdleTime>1200</MaxAppIdleTime>
  <RecordAppLog>false</RecordAppLog>
  <JSEngine>
    <RuntimeSize>1024</RuntimeSize>
    <MaxTimeOut>0</MaxTimeOut>
    <ScriptLibPath></ScriptLibPath>
  </JSEngine>
  <StreamManager>
    <StorageDir></StorageDir>
    <EnhancedSeek>false</EnhancedSeek>
    <KeyFrameInterval>1000</KeyFrameInterval>
  </StreamManager>
  <SharedObjManager>
    <StorageDir></StorageDir>
  </SharedObjManager>
  <Client>
    <Bandwidth override="yes">
      <ServerToClient>250000</ServerToClient>
      <ClientToServer>250000</ClientToServer>
    </Bandwidth>
    <BandwidthCap override="no">
      <ServerToClient>10000000</ServerToClient>
      <ClientToServer>10000000</ClientToServer>
    </BandwidthCap>
    <HTTPTunnel>
      <IdlePostInterval>512</IdlePostInterval>
      <IdleAckInterval>512</IdleAckInterval>
    </HTTPTunnel>
  </Client>
</Application>
```

The tags in the Application.xml file perform the following functions:

- `<Application>` is the root tag for the XML document. It is a container for all the other tags.
- `<LoadOnStartup>` indicates whether an application instance is loaded by default when the server starts up. Having an application instance loaded at server startup saves time when the first client connects to that application. The default value is `false`. If you set this tag to `true` in the virtual host's Application.xml file, an instance of each application on the server will be loaded at startup. If you set this tag to `true` in an application's optional Application.xml file, only that application will have an instance loaded at startup.
- `<MaxAppIdleTime>` indicates the number of seconds between the last client disconnection and the unloading of the application instance from the server's memory. The default is 1200 seconds (20 minutes). If this value is set to 0 or less, the default is used.
- `<RecordAppLog>` indicates whether to write the application log to a file. The default value is `false`. If you set this tag to `true`, a log stream file named *applicationName*.flv is created for each application in the virtual host's /admin/Streams/Logs directory. Each log file records connections to the application and any error messages that are generated by the server for the application.

  To view the contents of the log stream, you must build a communication application that retrieves the stream file from the server. Because the log files contain no audio or video, they are sent all at once by the server. For more information about creating communication applications, see *Developing Communication Applications.*

- `<JSEngine>` is a container for the next three tags, which control the resource usage of the Server-Side Communication ActionScript engine.
- `<RunTimeSize>` indicates the maximum number of bytes that a particular application instance can use to run server-side ActionScript on the server before garbage collection is performed—that is, before any unreferenced or unused ActionScript objects are purged from memory. The default is 1024K or 1 MB. If you create a client application that requires more than 1 MB of ActionScript memory, this value must be increased. If a new script object is created after garbage collection that will cause the runtime size of the application instance to exceed the value of the `<RunTimeSize>` tag, an out-of-memory error occurs and the application instance is shut down.
- `<MaxTimeOut>` indicates the maximum time in seconds that a server-side script function may take to execute. If a script function takes longer than the specified amount of time, it will be stopped by the server. Using this feature increases the processor workload. Setting this tag to a value greater than 0 is useful for debugging scripts during development of your applications. During deployment, it is recommended that you set this tag to the default value of 0, which imposes no limit on the time scripts take to execute.
- `<ScriptLibPath>` specifies a location for additional server-side scripts, such as components.asc or NetServices.asc. The server first looks for additional server-side scripts in the directory where your main.asc or *application_name*.asc file is located, which should be in /applications/*application_name* or /applications/*application_name*/scripts. Then, the server looks for additional scripts in the path specified in this tag. You can specify multiple directories by delimiting them with semicolons. Do not use quotation marks when specifying the path.
- `<StreamManager>` is a container for the next three tags, which control media streams.
- `<StorageDir>` indicates the directory where streams should be recorded for each application. The default is a directory named Streams in your application directory. The default is used if no directory is specified. When specifying a directory, use an absolute path.
- `<EnhancedSeek>` enables or disables finer seeking performance within streams. When this tag is set to `true`, the server inserts keyframes at the point in the stream where the seek begins if there is no preexisting keyframe there. This results in better visual display while seeking. When this tag is set to `false` (the default), no keyframes are inserted by the server and seeks begin at the nearest existing keyframe.
- `<KeyFrameInterval>` specifies, in milliseconds, how often to generate and save keyframes in an FLV file. These server-generated keyframes allow clients to perform seek and playback of the video stream. The default value is 1000.

  Setting this tag to a higher value than the default reduces the number of keyframes added to the FLV file and thus reduces the file size (but also reduces the seeking accuracy). For example, a 15-second video with a file size of 76 KB is increased only to 89 KB with `<KeyFrameInterval>` set to 5000, which is an increase of 13 KB, or 17%. The same video has a size of 109 KB with `<KeyFrameInterval>` set to 1000, which is an increase of 33 KB, or 43%. You should be aware of the correlation between file size and accuracy of seeking when you set this value.
- `<SharedObjManager>` is a container tag for the `<StorageDir>` tag.

- The <StorageDir> tag inside the <SharedObjManager> tag indicates the directory where shared objects should be stored. The default is a directory named Sharedobjects in the application's directory. The default is used if no directory is specified. When specifying a directory, use an absolute path. (For more information about shared objects, see *Developing Communication Applications*.)
- <Client> is a container for the <Bandwidth>, <BandwidthCap>, and <HTTPTunnel> tags.
- <Bandwidth> is a container for two tags that control the amount of bandwidth to use for upstream (client-to-server) and downstream (server-to-client) data traffic. By default, this tag includes an override parameter set to "yes", which allows both of its subtags to be overridden.
- <ServerToClient> specifies the maximum speed at which the server sends data to the client. The default is 250K (250,000 bytes per second) per client.
- <ClientToServer> specifies the maximum speed at which the client sends data to the server. The default is 250K (250,000 bytes per second) per client.
- <BandwidthCap> is a container for two tags that specify the maximum values that can be used when remotely editing the preceding <ServerToClient> and <ClientToServer> tags inside the <Bandwidth> tag. By default, this tag includes an override parameter set to "no". For more information about editing server configuration remotely, see the Macromedia Flash Support Center at www.macromedia.com/support/flash/.
- <ServerToClient> specifies the maximum speed that can be configured remotely for sending data to the client. The default is 10 MB (10,000,000 bytes per second).
- <ClientToServer> specifies the maximum speed that can be configured remotely for sending data to the server. The default is 10 MB (10,000,000 bytes per second).
- <HTTPTunnel> contains two tags that affect the latency observed by a client tunneling into the server (tunneling means to send RTMP packets through HTTP). Low values will reduce the latency but increase the network bandwidth overhead. The values of the two tags, <IdlePostInterval> and <IdleAckInterval>, should be considered together: for applications that need low latency, use a value of 128 for <IdlePostInterval> and 256 for <IdleAckInterval>; when latency is not an issue, use values of 1024 and 2048, respectively. The default setting of 512 for both values provides medium latency.
- <IdlePostInterval> specifies the interval, in milliseconds, at which the client should send idle posts to the server to indicate that the player has no data to send. This interval must be specified to enable the server to send data to the player. Valid values are between 0 and 4064 milliseconds.
- <IdleAckInterval> specifies the maximum number of millseconds the server waits before sending an acknowledgement (ack) of a client-idle post. Valid values are between 0 and 4064 milliseconds.

- The following set of Application.xml tags has been customized for real-world use. The `<RecordAppLog>` tag is set to `TRUE`. The runtime size of the JavaScript (JS) engine is increased from the default to 2 MB. The storage directory for streams is set to C:\FlashStreams\. The storage directory for shared objects is set to C:\FlashObjs\. The bandwidth limits are set to the defaults.

```
<Application>
  <LoadOnStartup>false</LoadOnStartup>
  <MaxAppIdleTime>1200</MaxAppIdleTime>
  <RecordAppLog>true</RecordAppLog>
  <JSEngine>
    <RuntimeSize>2048</RuntimeSize>
    <MaxTimeOut>0</MaxTimeOut>
    <ScriptLibPath>C:\Program Files\Macromedia Flash Communication Server
  MX\scriptlib</ScriptLibPath>
  </JSEngine>
  <StreamManager>
    <StorageDir>C:\FlashStreams\</StorageDir>
    <EnhancedSeek>false</EnhancedSeek>
    <KeyFrameInterval>5000</KeyFrameInterval>
  </StreamManager>
  <SharedObjManager>
    <StorageDir>C:\FlashObjs\</StorageDir>
  </SharedObjManager>
  <Client>
    <Bandwidth override= "yes">
      <ServerToClient>250000</ServerToClient>
      <ClientToServer>250000</ClientToServer>
    </Bandwidth>
    <BandwidthCap override= "no">
      <ServerToClient>10000000</ServerToClient>
      <ClientToServer>10000000</ClientToServer>
    </BandwidthCap>
    <HTTPTunnel>
      <IdlePostInterval>512</IdlePostInterval>
      <IdleAckInterval>512</IdleAckInterval>
    </HTTPTunnel>
  </Client>
</Application>
```

CHAPTER 4
Understanding Flash Communication Server Security

Macromedia Flash Communication Server MX 1.5 will typically be used in a network environment where many users will have access to it; by changing its configuration, you can make the server accessible from within a private network, from the public Internet, or both. When deploying any server technology, you will want to consider the implications to both the security of your internal network and the accessibility of the server's host computer.

Flash Communication Server incorporates security features that take these kinds of concerns into account. As a server administrator, you can provide additional security. This chapter describes the security features built into Flash Communication Server as well as additional measures you can take to protect your server.

Additional information about server security can be found in the Macromedia Flash Communication Server Support Center (www.macromedia.com/support/flashcom/).

Managing server security

Flash Communication Server uses a high-speed TCP/IP protocol called Real-Time Messaging Protocol (RTMP), which is binary and unencrypted. Because the protocol is unencrypted, you must carefully consider the security of your server configuration and the sensitivity of the data you send to and from the server.

Flash Communication Server's default settings at installation provide good security, but as a server administrator, you provide additional settings as you work with the configuration files and manage ongoing file deployment.

Edit the security tags in the XML files. Utilize the limits that can be set in the server's configuration files. Use the following tags in the XML files to enhance the server's security:

- Server.xml

  The <User> tags allow you to specify exactly who can connect to the server with the Administration Console. Only users specified with these tags can connect.

  The <Allow> and <Deny> tags let you specify exactly which domains administrators can connect from. Administrators cannot connect from domains that are not permitted with these tags. If you are running the server on a UNIX system, remember to allow connections from the domains where administrators will use the Administration Console to manage and monitor the server and its running applications.

  The <AdminServer>:<HostPort> tag allows you to specify the port of your choice for connecting to the Admin service with the Administration Console. This allows you to use a port that will work with your firewall configuration. The default is port 1111.

  The <ServerDomain> tag lets you specify the domain that the Flash Communication Server is running in so that it can identify its domain to application servers you may want it to connect to.

- Adaptor.xml

  The <ResourceLimits>:<MaxThreads> tag allows you to limit the number of threads to use for processing input/output requests. This prevents denial-of-service attacks from bringing down the server computer itself.

  The <Allow> and <Deny> tags let you specify exactly which domains administrators can connect from. Administrators cannot connect from domains that are not permitted with these tags. In the Adaptor.xml file, these tags indicate permissions specifically for the adaptor. If you are running the server on a UNIX system, remember to allow connections from the domains where administrators will use the Administration Console to manage and monitor the server and its running applications

  The <HostPort> tag allows you to specify the port to use for client connections. This lets you choose a port that works with your firewall configuration. The default for RTMP connections is port 1935.

- Vhost.xml

  The <ResourceLimits>:<MaxConnections> tag lets you specify the maximum number of simultaneous connections to allow on the virtual host. This can help denial-of-service attacks using very large numbers of connections. The default is -1, which allows unlimited connections.

  The <ResourceLimits>:<MaxAppInstances> lets you limit the number of application instances that can exist simultaneously on the virtual host. This can help prevent denial-of-service attacks. The default is -1, which allows unlimited application instances.

  The <ResourceLimits>:<MaxStreams> lets you specify the maximum number of streams that can exist simultaneously on the virtual host. This can help prevent denial of service attacks. The default is -1, which allows unlimited streams.

  The <ResourceLimits>:<MaxSharedObjects> lets you specify the maximum number of shared objects that can exist simultaneously on the virtual host. This can help prevent denial of service attacks. The default is -1, which allows unlimited shared objects.

  The <Allow> tag lets you limit connections to the virtual host to only clients who connect from the specified domains.

- Application.xml

  The <JSEngine>:<RuntimeSize> tag lets you limit the amount of memory that can be used by the server-side ActionScript on the virtual host. This can help prevent attacks using very large numbers of scripts. The default is 1024K.

  The <StreamManager>:<StorageDir> and <SharedObjManager>:<StorageDir> tags let you specify the locations for storing streams and shared objects. This allows you to store them in locations outside the applications directory in the Macromedia Flash Communication Server directory, if you wish.

  The <Bandwidth> tags groups let you specify the maximum amount of data that an application can send and receive.

For more detailed information about the server's XML tags, see Chapter 3, "Configuring Flash Communication Server," on page 27.

Place source and data files carefully. To prevent hackers from gaining access to the source files of your applications, avoid placing sensitive files in your web server's publishing directory. If you have a web server, the Administration Console (admin.swf) and sample applications are installed by default in your web server's publishing directory. During deployment, do not place Flash Communication Server application source or data files (FLA, FLV, ASC) or the applications directory (installed by default in the Flash Communication Server directory) in the web publishing directory; keep only your SWF and HTML files in the publishing directory.

Protect configuration files. In addition to its communication streams, the server's configuration files should be protected. To ensure that the server's XML files and directory structure cannot be accessed by unauthorized users, place the server computer in a physically secure location and password-protect the operating system so that only the appropriate server administrators have access.

About authentication and authorization

To authenticate (validate) administrators, Flash Communication Server employs several layers of host-based user security. (*Host-based security* refers to security measures that are implemented in the server software itself.) When a user tries to connect to the Administration Console with an administrator user name and password, the server uses the layers of settings in its XML configuration files to determine whether the connection should be allowed. Only administrators who have been explicitly defined can connect to the server to use the Administration Console.

The server authenticates administrators by evaluating the contents of the XML tags in the following order:

1. Server.xml file: <Allow>, <Deny>, and <User> tags. These tags indicate whether a user is allowed to connect to the Administration Console from the current IP address. Administrators can connect only from IP addresses you have specified with these tags.
2. Adaptor.xml file: <Allow> and <Deny> tags. These tags indicate whether a user is allowed to connect to the specified adaptor from the current IP address.
3. Vhost.xml file: <Allow> and <Deny> tags. These tags indicate whether a user is allowed to connect to the specified virtual host from the current IP address.

The server authenticates administrators by comparing their user names and passwords to those defined in the Server.xml file. When you choose these names and passwords, make sure they are not simple ones that can be easily guessed.

To have the server perform authentication of connecting users other than administrators, use the `<Allow>` and `<Deny>` tags in the Adaptor.xml and Vhost.xml files. With these tags you can prevent users from connecting from all domains other than those you specify. The server checks incoming connections against the Adaptor.xml file and then the Vhost.xml file when processing non-administrator connection requests.

To provide administrator *authorization* (assigning permissions), the server uses the Server.xml file. When you define a user as a server or virtual host administrator in this file, the server associates certain permissions with that user. Virtual host administrators can manage only a virtual host—for example, they can reload or disconnect applications on that virtual host. Server administrators can exercise control over all virtual hosts and perform server-level tasks, such as restarting or shutting down the server.

By default, only the Administration Console performs user authorization. When developing your own communication applications, you can decide whether to implement user authorization; some kinds of applications need this capability while others do not. For example, when developing a simple chat application, you might choose to create two different versions of your Macromedia Flash MX client movie. One version might be a chat participant version; another might be a chat moderator version, with additional functionality built in, such as the ability to edit users' posts or disconnect users. Using server-side ActionScript, you can define which users are able to connect with the moderator version of the movie.

As an additional security feature, the Administration Console actually connects to the Flash Communication Server Admin service, which then communicates with the server service to perform administration tasks.

Choosing passwords

When choosing passwords, remember to make them as secure as possible. The following guidelines can help you create more secure passwords:

- The minimum length of a password should be 7 characters.
- Passwords should not contain your user name or any part of it (for example: Jane, Doe, Jdoe).
- Passwords should contain three of the following four items: at least one uppercase letter (A-Z), at least one lowercase letter (a-z), at least one numeric character (0-9), and at least one non-alphanumeric character shown here:

  ! _ * # $ % & ; + -
- Passwords should be changed regularly and none of the last five passwords should be reused.

Developing secure applications

If you develop Flash Communication Server applications, you can use strategies to ensure the security of your applications and the data they use.

Confirm the location of the client SWF. When you deploy a Flash Communication Server application, use a server-side script to verify that connecting SWF files are coming from the location you expect (and not from an unknown computer). You can do this by checking the `client.referrer` property of the client object before the server accepts the connection.

For more information about writing server-side scripts, see *Developing Communication Applications*.

Use server-side script precautions. In server-side scripts do not use procedures that can be called by a malicious movie, which could then fill a hard disk, consume the processor, or do other damage. Procedures attached to client objects are particularly vulnerable. Procedures to be aware of include writing to the hard disk without checking the quantity of data being written, procedures that can be infinitely looped, and so on.

Send sensitive data via HTTPS. If you need to send sensitive data such as credit card information, you can use HTTPS to communicate simultaneously between your Flash MX client application and a separate application server that processes the data. To do this, use the ActionScript getURL command. (For more information, see the online ActionScript Dictionary in the Flash MX Help menu.)

About privacy

The technology in Macromedia Flash Communication Server MX enables the capture of client audio and video streams. When creating applications, it is your responsibility to comply with all applicable laws, rules, and regulations and to inform the user of privacy rights and your policies in situations such as when the application transports audio or video data across insecure channels or when audio or video data is being recorded for publication. For an example of adding user notification to your sample application, see "Adding a privacy module" in the "Application Development Tips and Tricks" chapter of *Developing Communication Applications*.

Deploying secure applications

When you deploy a Flash Communication Server application, it is important to take steps to ensure that your network is secure.

In addition to the precautions taken during the application development process, it is recommended that you deploy your communication applications in a firewall protected environment. Firewalls provide port-based protection for your network and can be used to prevent connections to the network from specific IP addresses.

You should take precautions when using log files to track server activity, since these files can consume large amounts of disk space over time.

The following two sections describe these precautions in more detail.

About firewalls

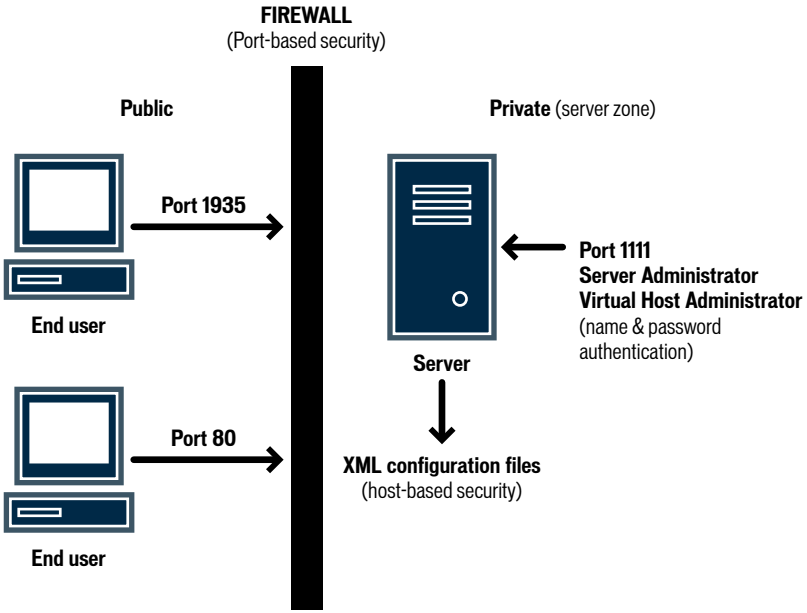
A *firewall* is a combination of hardware and software that controls the flow of information between networks, such as between a company intranet and the wider Internet. Firewalls provide port-based security, meaning they can be configured to allow certain communication ports (1935, 80) to appear "outside" the firewall, making them accessible to external networks.

The port that Flash Communication Server uses should be behind a firewall if it's being used only by users of a private network, such as a corporate intranet. The port should be accessible from outside the firewall if it's meant to be accessible to outside users such as users of the Internet in general.

If the Flash Communication Server and an application server are both behind a firewall, they can communicate with each other and no outside party can eavesdrop on the data to gain access to private information.

You can also configure a firewall to provide additional protection against outside attacks. For example, if the server is being flooded by a particular IP or range of IP addresses, you can configure the firewall to ignore messages from those IP addresses.

The server allows you to strictly control which users can connect to it and where they can connect from. You can also configure a firewall to control the ports users inside and outside your network can connect to.



Log file precautions

A log file is a file that contains information about events that have occurred on the server. When using log files (by specifying `true` for the Vhost.xml file's `<recordAccessLog>` tag and the Application.xml file's `<RecordAppLog>` tag), you are vulnerable to denial-of-service attacks by applications that can fill the hard disk—for example, by making high volumes of connection requests. To prevent this problem, write an operating system script to delete or back up the log regularly.

INDEX

A

Adaptor tag
  Adaptor.xml file 41
  Server.xml file 37
adaptors, configuring 41
Adaptor.xml file 41
  Adaptor tag 41
  Allow tag 42
  HostPort tag 42
  HostPortList tag 41
  MaxThreads tag 41
  ResourceLimits tag 41
Admin service 17
Admin tag 35
Admin Users panel 21
Administration Console 16
  Admin Users panel 21
  and administrator access 34, 52
  connecting 17, 36
  Diagnostics panel 18
  License panel 23
  Live Log panel 22
  Maintenance panel 19
administration tasks 11
  configuring adaptors 41
  configuring applications 13, 46
  configuring virtual hosts 13, 14, 19, 32, 37, 42
  defining administrators 21, 22, 36
  halting the server 20
  pinging the server 18
  reloading applications 19
  removing applications 19
  restarting the server 20
  restarting virtual hosts 20
  runtime configuration 24
  shutting down the server 20
  starting the server 15
  stopping the server 15
  stopping virtual hosts 20
  unloading applications 19
  uploading server-side scripts 14
  using the Windows event viewer 23
administrators
  defining 21, 36
  virtual hosts 13
AdminServer tag 38
Alias tag 43
AliasList tag 43
Allow tag
  Adaptor.xml file 42
  Server.xml file 36, 37
  Vhost.xml file 45
App inspector 15
AppInstanceGC tag 45
application server 29, 30, 31
Application tag 47
ApplicationGCtag (Server.xml file) 39
applications
  configuring 13, 46
  deploying server-side scripts 14
  registering 13
  removing 19
  unloading and reloading 19
applications directory 13
Application.xml file 12, 46
  Application tag 47
  Bandwidth tag 49
  BandwidthCap tag 49
  Client tag 49
  ClientToServer tag 49
  EnhancedSeek tag 48
  JSEngine tag 48
  LoadOnStartup tag 47
  MaxAppIdleTime tag 47
  MaxTimeOut tag 48
  RecordAppLog tag 47

RunTimeSize tag 48
ScriptLibPath tag 48
ServerToClient tag 49
SharedObjManager tag 48
StorageDir tag 48, 49
StreamManager tag 48
AppsDir tag 44
authentication 30, 31, 53
authorization 53

B

Bandwidth tag 49
BandwidthCap tag 49

C

client applications 12
configuring 13
server-side scripts 12
shared objects 12
streams 12
Client tag 49
ClientToServer tag 49
commands, on UNIX 24
Communication App inspector 15
configuration 27, 31, 34
of client applications 13
hierarchy 32
runtime 24
CPUMonitor tag (Server.xml file) 39

D

debugger, NetConnection 15
Deny tag 36, 38
deployment 28, 29, 30, 31
Diagnostics panel 18
DNSSuffix tag 45

E

EnhancedSeek tag 48
event viewer, Windows 23

F

fcsmgr utility 24
file permissions, on UNIX 38
firewalls 28, 55
FLVCacheSize tag (Server.xml file) 39

G

GID tag 38
for Admin service 39

H

halting the server 20
HostPort tag
Adaptor.xml file 42
Server.xml file 38
HostPortList tag 41
HTTP tunneling
and HostPort tag 42
and latency issues 49
HTTPS 55
HTTPTunnel tag 49

I

IdleAckInterval tag 49
IdlePostInterval tag 49
installation 7
files installed 8

J

JSEngine tag 48

L

License panel 23
LicenseInfo tag 38
Live Log panel 22
LoadOnStartup tag 47
log files 22, 44, 47, 56

M

MacSharedObjects tag 44
Maintenance panel 19
Mask tag 38
MaxAppIdleTime tag 47
MaxAppInstances tag 44
MaxConnections tag 44
MaxStreams tag 44
MaxThreads tag 41
MaxTimeOut tag 48

N

NetConnection Debugger 15

O

Order tag 37, 38

P

Password tag 36, 37
passwords 54
ping 18
platforms, supported 5
privacy 55

Process tag 38
for Admin service 39

R

RecordAccessLog tag 44
RecordAppLog tag 47
registering applications with the server 13
reloading applications 19
removing applications 19
ResourceLimits tag (Adaptor.xml file) 41
ResourceLimits tag (Server.xml file) 39
ResourceLimits tag (Vhost.xml file) 44
restarting the server 20
restarting virtual hosts 20
root 13
Root tag 35
runtime configuration 24
RunTimeSize tag 48

S

ScriptLibPath tag 48
security 51
authentication 30, 31, 53
authorization 53
file locations 53
file protection 53
firewalls 28, 55
HTTPS 55
log files 56
on UNIX systems 38
passwords 54
privacy 55
restricting connections 51, 54
server-side scripts 55
XML files 51
server
license 23
logs 22, 44, 47, 56
restarting 20
shutting down 20
Server tag 35
ServerDomain tag 39
server-side scripts 12, 14, 55
ServerToClient tag 49
Server.xml file 34
Adaptor tag 37
Admin tag 35
AdminServer tag 38
Allow tag 36, 37
ApplicationGC tag 39
CPUMonitor tag 39
Deny tag 36, 38
FLVCacheSize tag 39
HostPort tag 38
LicenseInfo tag 38
Order tag 37, 38
Password tag 36, 37
ResourceLimits tag 39
Root tag 35
Server tag 35
ServerDomain tag 39
ThreadPoolGC tag 39
User tag 36, 37
UserList tag 35, 37
VirtualHost tag 37
shared objects 12
SharedObjManager tag 48
shutting down the server 20
SocketGC tag 38
starting the server 15
stopping the server 15
stopping virtual hosts 20
StorageDir tag 48
StreamManager tag 48
streams 12
Streams tag 45
System requirements 5

T

ThreadPoolGC tag (Server.xml file) 39
tunneling, configuration tags for 49

U

UID tag 38
for Admin service 39
UNIX systems
commands for 24
fcsmgr 24
requirements for 6
UNIX systems, file permissions on 38
unloading applications 19
uptime 18
User tag 36, 37
UserList tag 35, 37

V

Vhost.xml file 42
Alias tag 43
AliasList tag 43
Allow tag 45
AppInstanceGC tag 45
AppsDir tag 44

DNSSuffix tag 45
MacSharedObjects tag 44
MaxAppInstances tag 44
MaxConnections tag 44
MaxStreams tag 44
RecordAccessLog tag 44
ResourceLimits tag 44
Streams tag 45
VirtualDirectory tag 45
VirtualHost tag 43
virtual hosts 19, 42
adding 32
administrators 13
configuring 13, 14, 19, 37, 42
restarting 20
starting 20
stopping 20
VirtualDirectory tag 45
VirtualHost tag
Server.xml file 37
Vhost.xml file 43

W

Windows event viewer 23